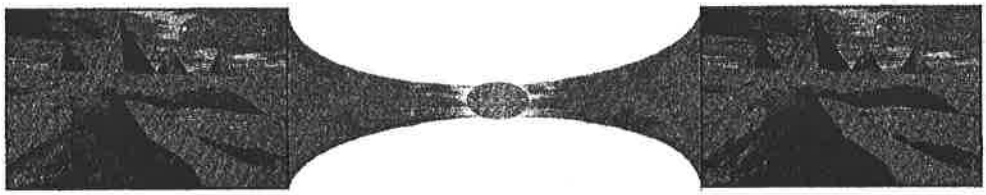
Exhibit Z

To

Joint Claim Chart

Rate-Distortion Optimization for VIDEO COMPRESSION



Gary J. Sullivan and Thomas Wiegand

The rate-distortion efficiency of today's video compression schemes is based on a sophisticated interaction between various motion representation possibilities, waveform coding of differences, and waveform coding of various refreshed regions. Hence, a key problem in high-compression video coding is the operational control of the encoder. This problem is compounded by the widely varying content and motion found in typical video sequences, necessitating the selection between different representation possibilities with varying rate-distortion efficiency. This article addresses the problem of video encoder optimization and discusses its consequences on the compression architecture of the overall coding system. Based on the well-known hybrid video coding structure, Lagrangian optimization techniques are presented that try to answer the question: "What part of the video signal should be coded using what method and parameter settings?"

Lagrangian methods for making good decisions in high-compression video coding

Video Compression Basics

Motion video data consists essentially of a time-ordered sequence of pictures, and cameras typically generate approximately 24, 25, or 30 pictures (or *frames*) per second. This results in a large amount of data that demands the use of compression. For example, assume that each picture has a relatively low "QCIF" (quarter-common-intermediate-format) resolution (i.e., 176×144 samples) for which each sample is digitally represented with 8 bits, and assume that we skip two out of every three pictures in order to cut down the bit rate. For color pictures, three color component samples are necessary to represent a sufficient color space for each pixel. In order to transmit even this relatively low-fidelity sequence of pictures, the raw source data rate is still more than 6 Mbit/s. However, today's low-cost transmission channels often operate at much lower data rates so that the data rate of the video signal needs to be further com-

1053-5888/98/$10.00©1998IEEE

A History of Existing Visual Coding Standards

H.120: The first international digital video coding standard [3]. It may have even been the first international digital compression standard for natural continuous-tone visual content of any kind (whether video or still picture). H.120 was developed by the ITU-T organization (the International Telecommunications Union—Telecommunications Standardization Sector, then called the CCITT), and received final approval in 1984. It originally was a conditional replenishment (CR) coder with differential pulse-code modulation (DPCM), scalar quantization, and variable-length coding, and it had an ability to switch to quincunx sub-sampling for bit-rate control. In 1988, a second version of H.120 added motion compensation and background prediction. (None of the later completed standards have yet included background prediction again, although a form of it is in the draft of the future MPEG-4 standard.) Its operational bit rates were 1544 and 2048 Kbit/s. H.120 is essentially no longer in use today, although a few H.120 systems are rumored to still be in operational condition.

H.261: The first widespread practical success—a video codec capable of operation at affordable telecom bit rates (with 80-320 Kbit/s devoted to video) [4, 5]. It was the first standard to use the basic typical structure we find still predominant today (16 × 16 macroblock motion compensation, 8 × 8 block DCT, scalar quantization, and two-dimensional run-level variable-length entropy coding). H.261 was approved by the ITU-T in early 1991 (with technical content completed in late 1990). It was later revised in 1993 to include a backward-compatible high-resolution graphics transfer mode. Its target bit-rate range was 64-2048 Kbit/s.

JPEG: A highly successful continuous-tone, still-picture coding standard named after the Joint Photographic Experts Group that developed it [1, 2]. Anyone who has browsed the world-wide web has experienced JPEG. JPEG (IS 10918-1/ITU-T T.81) was originally approved in 1992 and was developed as an official joint project of both the ISO/IEC JTC1 and ITU-T organizations. In its typical use, it is essentially H.261 INTRA coding with prediction of average values and an ability to customize the quantizer reconstruction scaling and the entropy coding to the specific picture content. However, there is much more in the JPEG standard than what is typically described or used. In particular, this includes progressive coding, lossless coding, and arithmetic coding.

MPEG-1: A widely successful video codec capable of approximately VHS videotape quality or better at about 1.5 Mbit/s and covering a bit rate range of about 1-2 Mbit/s [6, 7]. MPEG-1 gets its acronym from the Moving Pictures Experts Group that developed it [6, 7]. MPEG-1 video (IS 11172-2) was a project of the ISO/IEC JTC1 organization and was approved in 1993. In terms of technical features, it added bi-directionally predicted frames (known as B-frames) and half-pixel motion. (Half-pixel motion had been proposed during the development of H.261, but was apparently thought to be too complex at the time.) It provided superior quality than H.261 when operated at higher bit rates. (At bit rates below, perhaps, 1 Mbit/s, H.261 performs better, as MPEG-1 was not designed to be capable of operation in this range.)

MPEG-2: A step higher in bit rate, picture quality, and popularity. MPEG-2 forms the heart of broadcast-quality digital television for both standard-definition and high-definition television (SDTV and HDTV) [7-9]. MPEG-2 video (IS 13818-2/ITU-T H.262) was designed to encompass MPEG-1 and to also provide high quality with interlaced video sources at much higher bit rates. Although usually thought of as an ISO standard, MPEG-2 video was developed as an official joint project of both the ISO/IEC JTC1 and ITU-T organizations, and was completed in late 1994. Its primary new technical features were efficient handling of interlaced-scan pictures and hierarchical bit-usage scalability. Its target bit-rate range was approximately 4-30 Mbit/s.

H.263: The first codec designed specifically to handle very low-bit-rate video, and its performance in that arena is still state-of-the-art [10, 11]. H.263 is the current best standard for practical video telecommunication. Its original target bit-rate range was about 10-30 Kbit/s, but this was broadened during development to perhaps at least 10-2048 Kbit/s as it became apparent that it could be superior to H.261 at any bit rate. H.263 (version 1) was a project of the ITU-T and was approved in early 1996 (with technical content completed in 1995). The key new technical features of H.263 were variable block-size motion compensation, overlapped-block motion compensation (OBMC), picture-extrapolating motion vectors, three-dimensional run-level-last variable-length coding, median MV prediction, and more efficient header information signaling (and, relative to H.261, arithmetic coding, half-pixel motion, and bi-directional prediction—but the first of these three features was also found in JPEG and some form of the other two were in MPEG-1). At very low bit rates (e.g., below 30 Kbit/s), H.263 can code with the same quality as H.261 using half or less than half the bit rate [12]. At greater bit rates (e.g., above 80 Kbit/s) it can provide a more moderate degree of performance superiority over H.261. (See also H.263+ below.)

H.263+: Technically a second version of H.263 [10, 13]. The H.263+ project added a number of new optional features to H.263. One notable technical advance over prior standards is that H.263 version 2 was the first video coding standard to offer a high degree of error resilience for wireless or packet-based transport networks. H.263+ also added a number of improvements in compression efficiency, custom and flexible video formats, scalability, and backward-compatible supplemental enhancement information. It was approved in January of 1998 by the ITU-T (with technical content completed in September 1997). It extends the effective bit-rate range of H.263 to essentially any bit rate and any progressive-scan (noninterlace) picture formats and frame rates, and H.263+ is capable of superior performance relative to any existing standard over this entire range. The first author was the editor of H.263 during the H.263+ project and is the Rapporteur (chairman) of the ITU-T Advanced Video Coding Experts Group (SG16/Q15), which developed it.

The most successful class of video compression designs are called hybrid codecs.

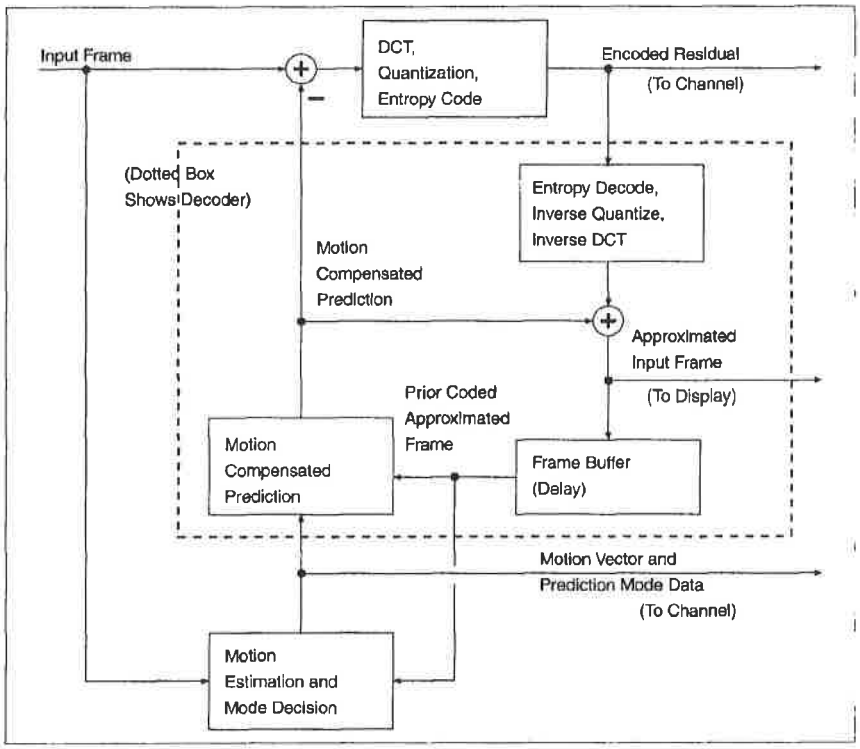
pressed. For instance, using V.34 modems that transmit at most 33.4 Kbit/s over dial-up analog phone lines, we still need to compress the video bit rate further by a factor of about 200 (more if audio is consuming 6 Kbit/s of that same channel or if the phone line is too noisy for achieving the full bit rate of V.34).

One way of compressing video content is simply to compress each picture, using an image-coding syntax such as JPEG [1, 2]. The most common "baseline" JPEG scheme consists of breaking up the image into equal-size blocks. These blocks are transformed by a discrete cosine transform (DCT), and the DCT coefficients are then quantized and transmitted using variable-length codes. We will refer to this kind of coding scheme as INTRA-frame coding, since the picture is coded without referring to other pictures in the video sequence. In fact, such INTRA coding alone (often called "motion JPEG") is in common use as a video coding method today in production-quality editing systems that demand rapid access to any frame of video content.

However, improved compression performance can be attained by taking advantage of the large amount of temporal redundancy in video content. We will refer to such techniques as INTER-frame coding. Usually, much of the depicted scene is essentially just repeated in picture after picture without any significant change. It should be obvious then that the video can be represented more efficiently by coding only the changes in the video content, rather than coding each entire picture repeatedly. This ability to use the temporal-domain redundancy to improve coding efficiency is what fundamentally distinguishes video compression from still-image compression.

A simple method of improving compression by coding only the changes in a video scene is called conditional replenishment (CR), and it was the only temporal redundancy reduction method used in the first digital video coding standard, ITU-T Rec. H.120 [3]. CR coding consists of sending signals to indicate which areas of a picture can just be repeated, and sending new coded information to replace the changed areas. CR thus allows a choice between one of two modes of representation for each area, which are called the SKIP mode and the INTRA mode. However, CR coding has a significant shortcoming, which is its inability to refine an approximation. Often the content of an area of a prior picture can be a good approximation of the new picture, needing only a minor alteration to become a better representation. But CR coding allows only exact repetition or complete replacement of each picture area. Adding a third type of "prediction mode," in which a refining *frame difference* approximation can be sent, results in a further improvement of compression performance.

The concept of frame difference refinement can also be taken a step further, by adding *motion-compensated* prediction (MCP). Most changes in video content are typically due to the motion of objects in the depicted scene relative to the imaging plane, and a small amount of motion can result in a large difference in the values of the pixels in a picture area (especially near the edges of an object). Often, *displacing* an area of the prior picture by a few pixels in spatial location can result in a significant reduction in the amount of information that needs to be sent as a frame difference approximation. This use of spatial displacement to form an approximation is known as motion com-



▲ *1. Typical motion-compensated DCT video coder.*

In practice, a number of interactions between coding decisions must be neglected in video coding optimization.

pensation and the encoder's search for the best spatial displacement approximation to use is known as motion estimation. The coding of the resulting difference signal for the refinement of the MCP signal is known as displaced frame difference (DFD) coding.

Hence, the most successful class of video compression designs are called hybrid codecs. The naming of this coder is due to its construction as a hybrid of motion-handling and picture-coding techniques, and the term codec is used to refer to both the coder and decoder of a video compression system. Figure 1 shows such a *hybrid* coder. Its design and operation involve the optimization of a number of decisions, including

1. How to segment each picture into areas,
2. Whether or not to replace each area of the picture with completely new INTRA-picture content,
3. If not replacing an area with new INTRA content
   (a) How to do motion estimation; i.e, how to select the spatial shifting displacement to use for INTER-picture predictive coding (with a zero-valued displacement being an important special case),
   (b) How to do DFD coding; i.e., how to select the approximation to use as a refinement of the INTER prediction (with a zero-valued approximation being an important special case), and
4. If replacing an area with new INTRA content, what approximation to send as the replacement content.

At this point, we have introduced a problem for the engineer who designs such a video coding system, which is: *What part of the image should be coded using what method*? If the possible modes of operation are restricted to INTRA coding and SKIP, the choice is relatively simple. However, hybrid video codecs achieve their compression performance by employing several modes of operation that are adaptively assigned to parts of the encoded picture, and there is a dependency between the effects of the motion estimation and DFD coding stages of INTER coding. The modes of operation are generally associated with signal-dependent rate-distortion characteristics, and rate-distortion trade-offs are inherent in the design of each of these aspects. The second and third items above in particular are unique to motion video coding. The optimization of these decisions in the design and operation of a video coder is the primary topic of this article. Some further techniques that go somewhat beyond this model will also be discussed.

Motion-Compensated Video Coding Analysis

Consider the nth coded picture of size $W \times H$ in a video sequence, consisting of an array $I_n(s)$ of color component values (e.g., $Y_n(s)$, $Cb_n(s)$, and $Cr_n(s)$) for each pixel location $s = (x, y)$, in which x and y are integers such that $0 \le x < W$ and $0 \le y < H$. The decoded approximation of this picture will be denoted as $\tilde{I}_n(s)$.

The typical video decoder (see Fig. 1) receives a representation of the picture that is segmented into some number K of distinct regional areas $\{\mathcal{A}_{i,n}\}_{i=1}^{K}$. For each area, a prediction-mode signal $p_{i,n} \in \{0,1\}$ is received indicating whether or not the area is predicted from the prior picture. For the areas that are predicted from the prior picture, a motion vector (MV), denoted $v_{i,n}$, is received. The MV specifies a spatial displacement for motion compensation of that region. Using the prediction mode and

An Overview of Future Visual Coding Standardization Projects

MPEG-4: A future visual coding standard for both still and moving visual content. The ISO/IEC SC29 WG11 organization is currently developing two drafts, called version 1 and version 2 of MPEG-4 visual. Final approval of version 1 is planned in January 1999 (with technical content completed in October 1998), and approval of version 2 is currently planned for approximately one year later. MPEG-4 visual (which will become IS 14496-2) will include most technical features of the prior video and still-picture coding standards, and will also include a number of new features such as zero-tree wavelet coding of still pictures, segmented shape coding of objects, and coding of hybrids of synthetic and natural video content. It will cover essentially all bit rates, picture formats, and frame rates, including both interlaced and progressive-scan video pictures. Its efficiency for predictive coding of normal camera-view video content will be similar to that of H.263 for noninterlaced video sources and similar to that of MPEG-2 for interlaced sources. For some special purpose and artificially generated scenes, it will provide significantly superior compression performance and new object-oriented capabilities. It will also contain a still-picture coder that has improved compression quality relative to JPEG at low bit rates.

H.263++: Future enhancements of H.263. The H.263++ project is considering adding more optional enhancements to H.263 and is currently scheduled for completion late in the year 2000. It is a project of the ITU-T Advanced Video Coding Experts Group (SG16/Q15).

JPEG-2000: A future new still-picture coding standard. JPEG-2000 is a joint project of the ITU-T SG8 and ISO/IEC JTC1 SC29 WG1 organizations. It is scheduled for completion late in the year 2000.

H.26L: A future new generation of video coding standard with improved efficiency, error resilience, and streaming support. H.26L is currently scheduled for approval in 2002. It is a project of the ITU-T Advanced Video Coding Experts Group (SG16/Q15).

MV, an MCP $\hat{I}_n(s)$ is formed for each pixel location $s \in \mathcal{A}_{i,n}$

$$\hat{I}_n(s) = p_{i,n} \cdot \hat{I}_{n-1}(s - v_{i,n}), s \in A_{i,n}. \quad (1)$$

(Note: The MV $v_{i,n}$ has no effect if $p_{i,n} = 0$ and so the MV is therefore normally not sent in that case.)

In addition to the prediction mode and MV information, the decoder receives an approximation $\tilde{R}_{i,n}(s)$ of the DFD residual error $R_{i,n}(s)$ between the true image value $I_n(s)$ and its MCP $\hat{I}_n(s)$. It then adds the residual signal to the prediction to form the final coded representation

$$\hat{I}_n(s) = \hat{I}_n(s) + \tilde{R}_{i,n}(s), s \in A_{i,n}. \quad (2)$$

Since there is often no movement in large parts of the picture, and since the representation of such regions in the previous picture may be adequate, video coders often provide special provisions for a SKIP mode of area treatment, which is efficiently transmitted using very short code words ($p_{i,n} = 1, v_{i,n} = 0, \tilde{R}_{i,n}(s) = 0$).

In video coders designed primarily for natural camera-view scene content, often little real freedom is given to the encoder for choosing the segmentation of the picture into region areas. Instead, the segmentation is typically either fixed to always consist of a particular two-dimensional block size (typically 16×16 pixels for prediction-mode signals and 8×8 for DFD residual content) or in some cases it is allowed to switch adaptively between block sizes (such as allowing the segmentation used for motion compensation to have either a 16×16 or 8×8 block size). This is because providing the encoder more freedom to specify a precise segmentation has generally not yet resulted in a significant improvement of compression performance for natural camera-view scene content (due to the number of bits needed to specify the segmentation), and also because determining the best possible segmentation in an encoder can be very complex. However, in special applications (especially those including artificially constructed picture content rather than camera-view scenes), segmented object-based coding can be justified. Rate-distortion optimization of segmentations for variable block-size video coding was first discussed in [30, 31], which was later enhanced to include dynamic programming to account for sequential dependencies in [37]-[39]. The optimization of coders that use object segmentation is discussed in an accompanying article [15].

Distortion Measures

Rate-distortion optimization requires an ability to measure distortion. However, the perceived distortion in visual content is a very difficult quantity to measure, as the characteristics of the human visual system are complex and not well understood. This problem is aggravated in video coding, because the addition of the temporal domain relative to still-picture coding further complicates

Standard Hybrid Video Codec Terminology

The following terms are useful for understanding the various international standards for video coding:

prediction mode: A basic representation model that is selected for use in approximating a picture region (INTRA, INTER, etc.).

mode decision: An encoding process that selects the prediction mode for each region to be encoded.

block: A rectangular region (normally of size 8×8) in a picture. The discrete cosine transform (DCT) in standard video coders operates on 8×8 block regions.

macroblock: A region of size 16×16 in the luminance picture and the corresponding region of chrominance information (often an 8×8 region), which is associated with a prediction mode.

motion vector (MV): A spatial displacement offset for use in the prediction of an image region. In the INTER prediction mode an MV affects a macroblock region, while in the INTER+4V prediction mode, an individual MV is sent for each of the four 8×8 luminance blocks in a macroblock.

motion compensation: A decoding process that represents motion in each region of a picture by application of the transmitted MVs to the prior decoded picture.

motion estimation: An encoding process that selects the MVs to be used for motion compensation.

half-pixel motion: A representation of motion in which an MV may specify prediction from pixel locations that are halfway between the pixel grid locations in the prior picture, thus requiring interpolation to construct the prediction of an image region.

picture-extrapolating MVs: A representation of motion in which an MV may specify prediction from pixel locations that lie partly or entirely outside the boundaries of the prior picture, thus requiring extrapolation of the edges of the picture to construct the prediction of an image region.

overlapped-block motion compensation (OBMC): A representation of motion in which the MVs that represent the motion in a picture have overlapping areas of influence.

INTRA mode: A prediction mode in which the picture content of a macroblock region is represented without reference to a region in any previously decoded picture.

SKIP mode: A prediction mode in which the picture content of a macroblock region is represented as a copy of the macroblock in the same location in a previously decoded picture.

INTER mode: A prediction mode in which the picture content of a macroblock region is represented as the sum of a motion-compensated prediction using a motion vector, plus (optionally) a decoded residual difference signal representation.

INTER+4V mode: A prediction mode in which the picture content of a macroblock region is represented as in the INTER mode, but using four motion vectors (one for each 8×8 block in the macroblock).

INTER+Q mode: A prediction mode in which the picture content of a macroblock is represented as in the INTER mode, and a change is indicated for the inverse quantization scaling of the decoded residual signal representation.

the issue. In practice, highly imperfect distortion models such as the sum of squared differences (SSD) or its equivalents, known as mean squared error (MSE) or peak signal-to-noise ratio (PSNR), are used in most actual comparisons. They are defined by

$$\mathrm{SSD}_{\mathcal{A}}(F,G) = \sum_{s\in\mathcal{A}} |F(s) - G(s)|^2 \tag{3}$$

$$\mathrm{MSE}_{\mathcal{A}}(F,G) = \frac{1}{|\mathcal{A}|}\,\mathrm{SSD}_{\mathcal{A}}(F,G) \tag{4}$$

$$\mathrm{PSNR}_{\mathcal{A}}(F,G) = 10\log_{10}\frac{(255)^2}{\mathrm{MSE}_{\mathcal{A}}(F,G)}\ \text{decibels}. \tag{5}$$

Another distortion measure in common use (since it is often easier to compute) is the sum of absolute differences (SAD)

$$\mathrm{SAD}_{\mathcal{A}}(F,G) = \sum_{s\in\mathcal{A}} |F(s) - G(s)| \tag{6}$$

where F and G are two array arguments (such as luminance arrays of the actual and approximated pictures). These measures are often applied to only the luminance field of the picture during optimization processes, but better performance can be obtained by including all three color components. (The chrominance components are often treated as something of a minor nuisance in video coding; since they need only about 10% of the bit rate of the luminance they provide a limited opportunity for optimization gain.)

Effectiveness of Basic Technical Features

In the previous sections we described the various technical features of a basic modern video coder. The effectiveness of these features and the dependence of this effectiveness on video content is shown in Fig. 2. The upper plot of Fig. 2 shows performance for a videophone sequence known as *Mother & Daughter*, with moderate object motion and a stationary background. The lower plot of Fig. 2 shows performance for a more demanding scene known as *Foreman*, with heavy object motion and an unstable hand-held moving camera. Each sequence was encoded in QCIF resolution at 10 frames per second using the framework of a well-optimized H.263 [10] video encoder (using optimization methods described later in this article). (H.263 has 16×16 prediction-mode regions called macroblocks and 8×8 DCT-based DFD coding.)

Complicating Factors in Video Coding Optimization

The video coder model described in this article is useful for illustration purposes, but in practice actual video coder designs often differ from it in various ways that complicate design and analysis. Some of the important differences are described in the following few paragraphs.

Color chrominance components (e.g., $Cb_n(s)$ and $Cr_n(s)$) are often represented with lower resolution (e.g., $W/2 \times H/2$) than the luminance component of the image $Y(s)$. This is because the human psycho-visual system is much more sensitive to brightness than to chrominance, allowing bit-rate savings by coding the chrominance at lower resolution. In such a system, the method of operation must be adjusted to account for the difference in resolution (for example, by dividing the MV values by two for chrominance components).

Since image values $I_n(s)$ are defined only for integer pixel locations $s = (x, y)$ within the rectangular picture area, the above model will work properly in the strict sense only if every motion vector $v_{i,n}$ is restricted to have an integer value and only a value that causes access to locations in the prior picture that are within the picture's rectangular boundary. These restrictions, which are maintained in some early video-coding methods such as ITU-T Rec. H.261 [4], are detrimental to performance. More recent designs such as ITU-T Rec. H.263 [10] support the removal of these restrictions by using interpolation of the prior picture for any fractional-valued MVs (normally half-integer values, resulting in what is called half-pixel motion) and MVs that access locations outside the boundary of the picture (resulting in what we call picture-extrapolating MVs). The prediction of an image area may also be filtered to avoid high-frequency artifacts (as in Rec. H.261 [4]).

Often there are interactions between the coding of different regions in a video coder. The number of bits needed to specify an MV value may depend on the values of the MVs in neighboring regions. The areas of influence of different MVs can be overlapping due to overlapped-block motion compensation (OBMC) [16]-[19], and the areas of influence of coded transform blocks can also overlap due to the application of deblocking filters. While these cross-dependencies can improve coding performance, they can also complicate the task of optimizing the decisions made in an encoder. For this reason these cross-dependencies are often neglected (or only partially accounted for) during encoder optimization.

One important and often-neglected interaction between the coding of video regions is the temporal propagation of error. The fidelity of each area of a particular picture will affect the ability to use that picture area for the prediction of subsequent pictures. Real-time encoders must neglect to account for this aspect to a large extent, since they cannot tolerate the delay necessary for optimizing a long temporal sequence of decisions with accounting for the temporal effects on many pictures. However, even nonreal-time encoders also often neglect this to account for this propagation in any significant way, due to the sheer complexity of adding this extra dimension to the analysis. An example for the exploitation of temporal dependencies in video coding can be found in [20]. The work of Ramchandran, Ortega, and Vetterli in [20] was extended by Lee and Dickinson in [21].

A gain in performance is shown for forming a CR coder by adding the SKIP coding mode to the encoder. Further gains in performance are shown when adding the various INTER coding modes to the encoder that were discussed in the previous sections:

▲ INTER (MV = (0,0) only): frame-difference coding with only zero-valued MV displacements

▲ INTER (Full-pixel motion compensation): integer-pixel (full-pixel) precision motion compensation with DFD coding

▲ INTER (Half-pixel motion compensation): half-pixel precision motion compensation with DFD coding

▲ INTER & INTER+4V: half-pixel precision motion compensation with DFD coding and the addition of an "advanced prediction" mode (H.263 Annex F), which includes a segmentation switch allowing a choice of either one or four MVs per 16 × 16 area and also includes overlapped-block motion compensation (OBMC) and picture-extrapolating MVs [10]. (The use of four MVs per macroblock is called the INTER+4V prediction mode.)

Except in the final case, the same H.263 baseline syntax was used throughout, with changes only in the coding method (the lower four curves are thus slightly penalized in performance by providing syntactical support for features that are never used in the encoding). In the final case, H.263 syntax was used with its D and F annexes active [10].

However, the coding results for the two sequences differ. In the low-motion sequence, the gain achieved by using CR (a choice of SKIP or INTRA) instead of just INTRA-picture coding is the most substantial, and as more features are added, the benefits diminish. On the high-motion sequence, CR is not very useful because the whole picture is changing from frame to frame, and the addition of motion compensation using the various INTER modes provides the most significant gain, with further gain added by each increasing degree of sophistication in motion handling.

Optimization Techniques

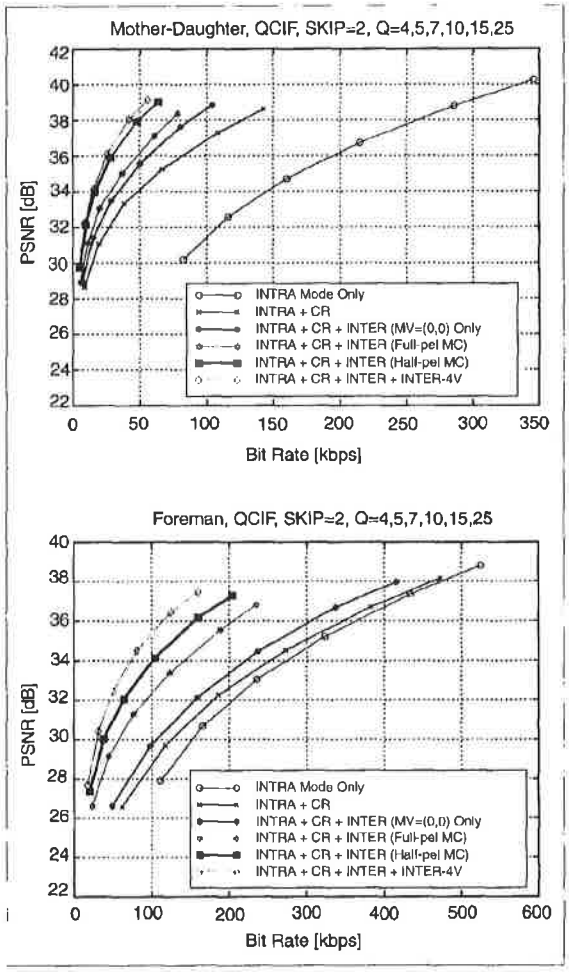
In the previous section, it was demonstrated that by adding efficient coding options in the rate-distortion sense to a video codec, the overall performance increases. The optimization task is to choose, for each image region, the most efficient coded representation (segmentation, prediction modes, MVs, quantization levels, etc.) in the rate-distortion sense. This task is complicated by the fact that the various coding options show varying efficiency at different bit rates (or levels of fidelity) and with different scene content.

For example, in H.263 [10], block-based motion compensation followed by quantization of the prediction error (INTER mode) is an efficient means for coding much of the key changing content in image sequences. On the other hand, coding a particular macroblock directly (INTRA mode) may be more productive in situations when the block-based translational motion model breaks down. For relatively dormant regions of the video, simply copying a portion of the previously decoded frame into the current frame may be preferred (SKIP mode). Intuitively, by allowing multiple modes of operation, we expect improved rate-distortion performance if the modes can significantly customize the coding for different types of scene statistics, and especially if the modes can be applied judiciously to different spatial and temporal regions of an image sequence.

The modes of operation that are assigned to the image regions have differing rate-distortion characteristics, and the goal of an encoder is to optimize its overall fidelity: *Minimize distortion D, subject to a constraint R_c on the number of bits used R.* This constrained problem reads as follows

$$\min\{D\}, \quad \text{subject to } R < R_c. \tag{7}$$

The optimization task in Eq. (7) can be elegantly solved using Lagrangian optimization where a distortion term is



▲ *2. Coding performance for the sequences Mother & Daughter (top) and Foreman (bottom).*

weighted against a rate term [22-32]. The Lagrangian formulation of the minimization problem is given by

$$\min\{J\}, \quad \text{where} \quad J = D + \lambda R, \tag{8}$$

where the Lagrangian rate-distortion functional J is minimized for a particular value of the Lagrange multiplier λ. Each solution to Eq. (8) for a given value of the Lagrange multiplier λ corresponds to an optimal solution to Eq. (7) for a particular value of R_c [22, 23]. More details on Lagrangian optimization are discussed in the accompanying article by Ortega and Ramchandran [14].

This technique has gained importance due to its effectiveness, conceptual simplicity, and its ability to effectively evaluate a large number of possible coding choices in an optimized fashion. If Lagrangian bit allocation and entropy codes for signaling the coding modes are used, the number of choices available for use need not be restricted to just a few. As a result, the computation time to test all the modes may become the limiting factor on performance, rather than the capabilities of the syntax itself.

In practice, a number of interactions between coding decisions must be neglected in video coding optimization. The primary problems are the use of motion estimation and prediction mode decisions, and the common presence of cascading effects of decisions made for one region on the coding of subsequent regions in space and time. In addition, the overall bit rate must typically be controlled to match the channel capacity—which further complicates matters. All three quantities, D, λ, and R, tend to be subject to approximations and compromises in designing video coding systems.

Bit-Rate Control

The overall bit rate of a video coder is determined by its prediction-mode decisions, MV choices, and DFD coding fidelity. The last of these three is typically the most important for bit-rate control, and the residual fidelity is typically controlled by choosing a step-size scaling to be used for inverse quantization reconstruction of the transformed difference signal [33]. A larger step size results in a lower bit rate and a larger amount of distortion. Thus, the choice of step size is closely related to the choice of the relative emphasis to be placed on rate and distortion; i.e., the choice of λ. (The choice of the quantizer step-size scaling must be communicated to the decoder, but λ is an encoder-only issue and is not needed by the decoder.) As a last resort, the coding of entire pictures can be skipped by the encoder as a bit-rate control mechanism (resulting in a less fluid rendition of motion).

In some cases the bit rate must be controlled to maintain a constant local-average bit rate over time, but in other cases it may be allowed to vary much more widely (such as by allowing the amount of scene content activity to govern the bit rate). Whatever the constraints imposed on the bit rate of the system, control over λ in a well-optimized encoder can provide an excellent means of meeting those constraints. In a later section we will show how control over λ can be tightly linked to the more conventional practice of control over the inverse quantization step size.

Lagrange multiplier optimization can provide significant benefits if judiciously applied.

A feedback control of the buffer state of video codecs was proposed by Choi and Park in [34], where the control is applied to the Lagrange multiplier λ. Trellis-based buffer control has been presented by Ortega, Ramchandran, and Vetterli in [35], where fast approximations are achieved using the Lagrangian formulation. A low-delay rate control method for H.263 was provided in [36]. There are many approaches to rate control; However, the use of the Lagrange multiplier method of optimization within these rate-control schemes can often help to avoid losses in coding performance that might otherwise result from their use.

Motion Estimation

Ideally, decisions should be controlled by their ultimate effect on the resulting pictures; however, this ideal may not be attainable in an encoder implementation. For example, in considering each possible MV to send for a picture area, an encoder should perform an optimized coding of the residual error and measure the resulting bit usage and distortion. Only by doing this can it really choose the best possible MV value to send (even if neglecting the effect of that choice on later choices spatially and later pictures temporally). However, there are typically thousands of possible MV values to choose from, and coding just one residual difference signal typically requires a significant fraction of the total computational power of a practical encoder.

A simpler method of performing motion estimation is to simply search for an MV that minimizes the prediction error prior to residual coding, perhaps giving some special preference to the zero-valued MV and to the MV value that requires the fewest bits to represent as a result of MV prediction in the decoder. These biases prevent spurious large MV values (which require a large number of bits to represent but may provide only little prediction benefit).

Further simplification is needed in real-time implementations. A straightforward minimum-squared-error "full-search" motion estimation that tests all possible integer values of an MV within a $\pm L$ range (video coding syntax typically supports $L = 16$ or $L = 32$, and one optional mode of H.263 supports an unlimited range) would require approximately $3(2L+1)^2$ operations per pixel (two adds and one multiply per tested MV value).

Adding half-pixel MVs to the search multiplies the number of MV values to test by a factor of four, and adds the requirement of an interpolation operation for generating the half-pixel sampling-grid locations in the prior picture. Such levels of complexity are beyond the capabilities of many of today's video coder implementations—and if this much computational power was available to an implementation, devoting it all to this type of search might not be the best way to gain performance. Motion estimation complexity is often reduced in implementations by the use of iterative refinement techniques. While we do not specifically address reduced-complexity motion estimation herein, rate-distortion optimization within the context of a reduced-complexity search can also often provide a performance benefit.

We can view MCP formation as a source coding problem with a fidelity criterion, closely related to vector quantization. For the number of bits required to transmit the MVs, MCP provides a version of the video signal with a certain fidelity. The rate-distortion trade-off can be controlled by various means. One approach is to treat MCP as *entropy-constrained vector quantization* (ECVQ) [24, 31]. Here, each image block to be encoded is quantized using its own codebook that consists of a neighborhood of image blocks of the same size in the previously decoded frames (as determined by the motion estimation search range). A codebook entry is addressed by the translational MVs, which are entropy coded. The criterion for the block motion estimation is the minimization of a Lagrangian cost function wherein the distortion, represented as the prediction error in SSD or SAD, is weighted against the number of bits associated with the translational MVs using a Lagrange multiplier.

An alternative interpretation is to view the motion search as an estimation problem: the estimation of a motion displacement field for the image. The problem of motion estimation becomes increasingly ill-conditioned as we increase the motion estimation search range and reduce the block size. The ill-conditioning results in a lack of consistency in the estimated MVs, resulting in a loss of accuracy in estimating true motion. The Lagrangian formulation can regularize the displacement field estimate. Hence, the Lagrangian formulation yields a solution to the problem not only when viewing motion estimation as a source coding technique, but also when viewing it an ill-conditioned displacement field estimation problem.

Block motion estimation can therefore be viewed as the minimization of the Lagrangian cost function

$$J_{\text{MOTION}} = D_{\text{DFD}} + \lambda_{\text{MOTION}} R_{\text{MOTION}}, \tag{9}$$

in which the distortion D_{DFD}, representing the prediction error measured as SSD or SAD, is weighted against the number of bits R_{MOTION} associated with the MVs using a Lagrange multiplier λ_{MOTION}. The Lagrange multiplier imposes the rate constraint as in ECVQ, and its value directly controls the rate-distortion trade-off, meaning that small values of λ_{MOTION} correspond to high fidelities and bit rates and large values of λ_{MOTION} correspond to lower fidelities and bit rates. Sullivan and Baker proposed such a rate-distortion-optimized motion estimation scheme for fixed or variable block sizes in [31], and more work on the subject has appeared in [32] and [37]-[42].

Variable Block Sizes

The impact of the block size on MCP fidelity and bit rate are illustrated in Fig. 3 for the video sequences *Mother & Daughter* (top) and *Foreman* (bottom). For the data in this figure, the motion estimation and compensation were performed using the sequence of original video frames, with temporal subsampling by a factor of 3. The motion estimation was performed by minimizing J_{MOTION} in Eq. (9). In the first part of the motion estimation procedure, an integer-pixel-accurate displacement vector was found within a search range of $[-15..15] \times [-15..15]$ pixels relative to the location of the block to be searched. Then, given this integer-pixel-accurate displacement vector, its surrounding half-pixel positions were checked for improvements when evaluating Eq. (9). This second stage of this process is commonly called half-pixel refinement.

For the curves in Fig. 3, we tested the impact of motion compensation block size on coding performance. For this test we evaluated three different sets of choices for 16×16 macroblock prediction modes:

- ▲ Case 1: INTER-coding using only luminance regions of size 16×16 samples (choosing between the SKIP mode signaled with codeword "1" and the INTER mode signaled with codeword "0" followed by a MV for the 16×16 region)
- ▲ Case 2: INTER-coding using only luminance blocks of size 8×8 samples (choosing between the SKIP mode signaled with codeword "1," the INTER+4V mode signaled with codeword "0" followed by four MVs for 8×8 regions)
- ▲ Case 3: Combining cases 1 and 2 using a rate-constrained encoding strategy, which adapts the frequency of using the various region sizes using Lagrange multiplier optimization (choosing between "1" for SKIP, "01" for INTER with a MV for the 16×16 region, and "00" for INTER+4V with four MVs for 8×8 regions) [31].

Each MV was represented using the H.263 method of MV prediction and variable-length coding.

Case 1 can achieve better prediction than case 2 at the lowest mode-decision bit rates, since it can represent a moving area with one fourth as many MVs. However, case 1 cannot achieve the prediction quality of case 2 when the bit rate is higher, because case 2 can represent finer motion detail. Summarizing the comparison of cases 1 and 2, the use of 16×16 blocks is more beneficial at low rates, while 8×8 blocks are desirable at high rates. Case 3 can adaptively choose the proper block size as needed, so it obtains the best prediction at virtually all bit rates. (Case 1 has better prediction than case 3 at the very lowest mode-decision bit rates, since it does not require

an extra bit per non-SKIP macroblock to distinguish between the INTER and INTER+4V cases.)

The ultimate impact of the block size on the final objective coding fidelity is shown in Fig. 4. In this experiment, the residual coding stage and the INTRA coding mode were added to the scheme, producing a complete encoder. A complete H.263 video coder (using annexes D and F) was used for this experiment. The tendencies observed for the case of motion compensation only (see Figure 3) are also true here. Allowing a large range of coding fidelities for MCP provides superior performance over the entire range of bit rates. However, every extra bit spent for motion compensation must be justified against other coding decisions made in mode selection and residual coding [32].

Other Methods for Improving Motion-Compensated Prediction

Besides block size variation to improve the MCP, various other methods have been proposed. Examples of these schemes include

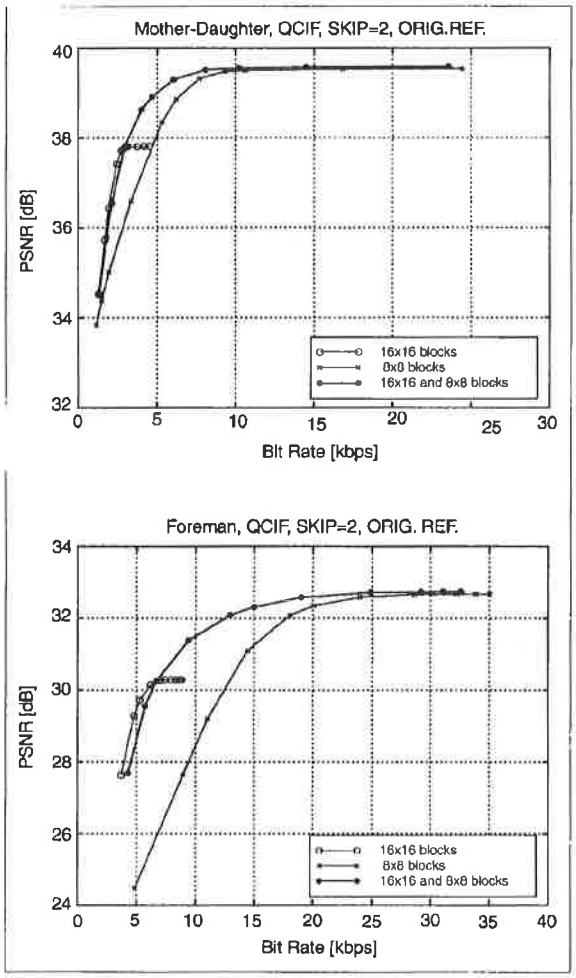
1. Multi-hypothesis MCP
2. Long-term memory MCP
3. Complex motion models

The scheme used for the first item, multi-hypothesis MCP, is that various signals are superimposed to compute the MCP signal. The multi-hypothesis motion-compensated predictor for a pixel location $s \in \mathcal{A}_{i,n}$ is defined as

$$\hat{I}_n(s) = \sum_{p=1}^{P} h_p(s) \cdot \hat{I}_{n-\Delta n}(s - v_{i,n,p}), s \in \mathcal{A}_{i,n}. \quad (10)$$

with $\hat{I}_n(s)$ being a predicted pixel value and $\hat{I}_{n-\Delta n}(s - v_{i,n,p})$ being a motion-compensated pixel from a decoded frame Δn time instants in the past (normally $\Delta n = 1$). This scheme is a generalization of Eq. (1) and it includes concepts like subpixel accurate MCP [43, 44], B-frames [45], spatial filtering [4], and OBMC [16-19]. Using the linear filtering approach of Eq. (10), the accuracy of motion compensation can be significantly improved. A rationale for this approach is that if there are P different plausible hypotheses for the MV that properly represents the motion of a pixel s, and if each of these can be associated with a hypothesis probability $h_p(s)$, then the expected value of the pixel prediction is given by Eq. (10)—and an expected value is the estimator that minimizes the mean-square error in the prediction of any random variable. Another rationale is that if each hypothesis prediction is viewed as a noisy representation of the pixel, then performing an optimized weighted averaging of the results of several hypotheses as performed in Eq. (10) can reduce the noise. It should be obvious that if an optimized set of weights $\{h_p(s)\}_{p=1}^{P}$ is used in the linear combination (Eq. (10)), the result cannot be worse on average than the result obtained from a single hypothesis as in Eq. (1). The multi-hypothesis MCP concept was introduced in [18], and an estimation-theoretic analysis with a focus on OBMC was presented in [19]. A rate-distortion efficiency analysis including OBMC and B-frames was presented in [46], and an algorithm to take advantage of MCP in an entropy-constrained framework was proposed in [47, 48].

The second item, long-term memory MCP as proposed in [49], refers to extending the spatial displacement vector utilized in block-based hybrid video coding by a variable time delay, permitting the use of more frames than the last prior decoded one for MCP. The long-term memory covers several seconds of decoded frames at the encoder and decoder. Experiments when employing 50 frames for motion compensation using the sequences *Foreman* and *Mother & Daughter* show that long-term memory MCP yields about 2 dB and 1 dB PSNR improvements in prediction error against the one-frame case, respectively. However, in the long-term-memory



▲ *3. Prediction gain vs. MV bit rate for the sequences Mother & Daughter (top) and Foreman (bottom) when employing H.263 MV median prediction and original frames as reference frames.*

case, the portion of the bit rate used for sending MVs shows an increase of 30% compared to the one-frame case [49]. Embedded in a complete video coder, the approach still yields significant coding gains expressed in bit-rate savings of 23% for the sequence *Foreman* and 17% for the sequence *Mother & Daughter* due to the impact of long-term memory MCP when comparing it to the rate-distortion optimized H.263 coder, which is outlined in this article [49].

Complex motion models (the third item) have been proposed by a great number of researchers for improving motion compensation performance. The main effect of using a higher-order approximation of the displacement vector field (e.g., using polynomial motion models) is increased accuracy relative to what is achievable with translational motion models that relate to piecewise-constant approximation. In [50] and [51], a complete video codec is presented, where image segments are motion compensated using bilinear (12 parameter) motion models. The image segments partition a video frame down to a granularity of 8×8 blocks. Bit-rate savings of more than 25% were reported for the sequence *Foreman* [51].

INTRA/INTER/SKIP Mode Decision

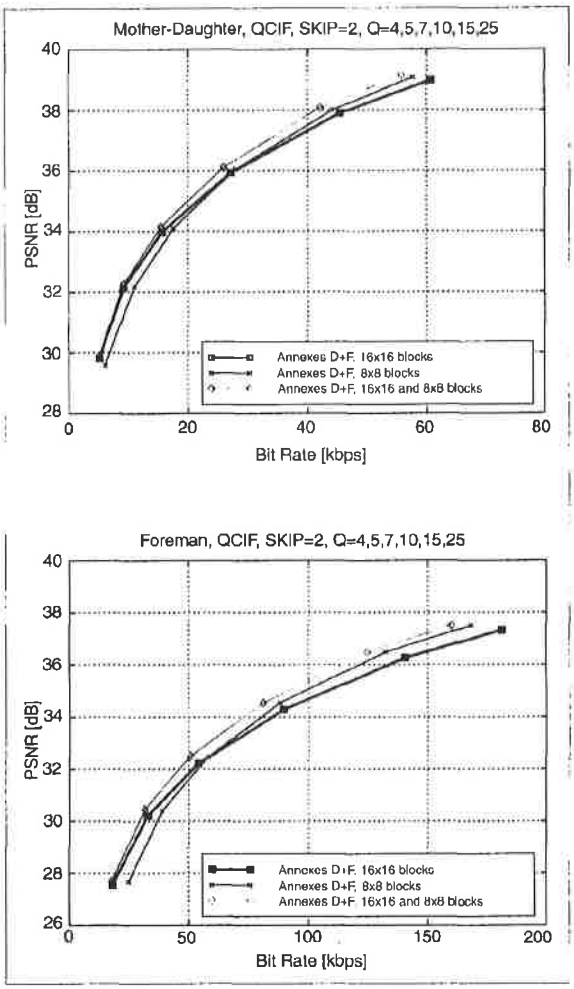
Hybrid video coding consists of the motion estimation and the residual coding stages, and an interface between them consisting of prediction mode decision. The task for the residual coding is to represent signal parts that are not sufficiently approximated by the earlier stages. From the viewpoint of bit-allocation strategies, the various prediction modes relate to various bit-rate partitions. Considering the various H.263 modes: INTRA, SKIP, INTER, and INTER+4V, Table 1 gives typical values for the bit-rate partition of motion and DFD texture coding for typical sequences. The various modes in Table 1 relate to quite different overall bit rates. Since the choice of mode is adapted to the scene content, it is transmitted as side information.

If we assume for the moment that the bit rate and distortion of the residual coding stage is controlled by the selection of a quantizer step size Q, then rate-distortion optimized mode decision refers to the minimization of the following Lagrangian functional

$$J(\mathcal{A}, M, Q) = D_{\text{REC}}(\mathcal{A}, M, Q) + \lambda_{\text{MODE}} R_{\text{REC}}(\mathcal{A}, M, Q), \quad (11)$$

where, for instance, $M \in \{\text{INTRA, SKIP, INTER, INTER+4V}\}$ indicates a mode chosen for a particular macroblock, Q is the selected quantizer step size, $D_{\text{REC}}(\mathcal{A}, M, Q)$ is the SSD between the original macroblock $\mathcal{A}$ and its reconstruction, and $R_{\text{REC}}(\mathcal{A}, M, Q)$ is the number of bits associated with choosing M and Q.

A simple algorithm for rate-constrained mode decision minimizes Eq. (11) given all mode decisions of past macroblocks [26, 27]. This procedure partially neglects dependencies between macroblocks, such as prediction of MV values from those of neighboring blocks and OBMC. In [52, 53], Wiegand et al. proposed the exploitation of mode decision dependencies between macroblocks using dynamic programming methods. Later work on the subject that also included the option to change the quantizer



▲ *4. Coding performance for the sequences Mother & Daughter (top) and Foreman (bottom) when employing variable block sizes.*

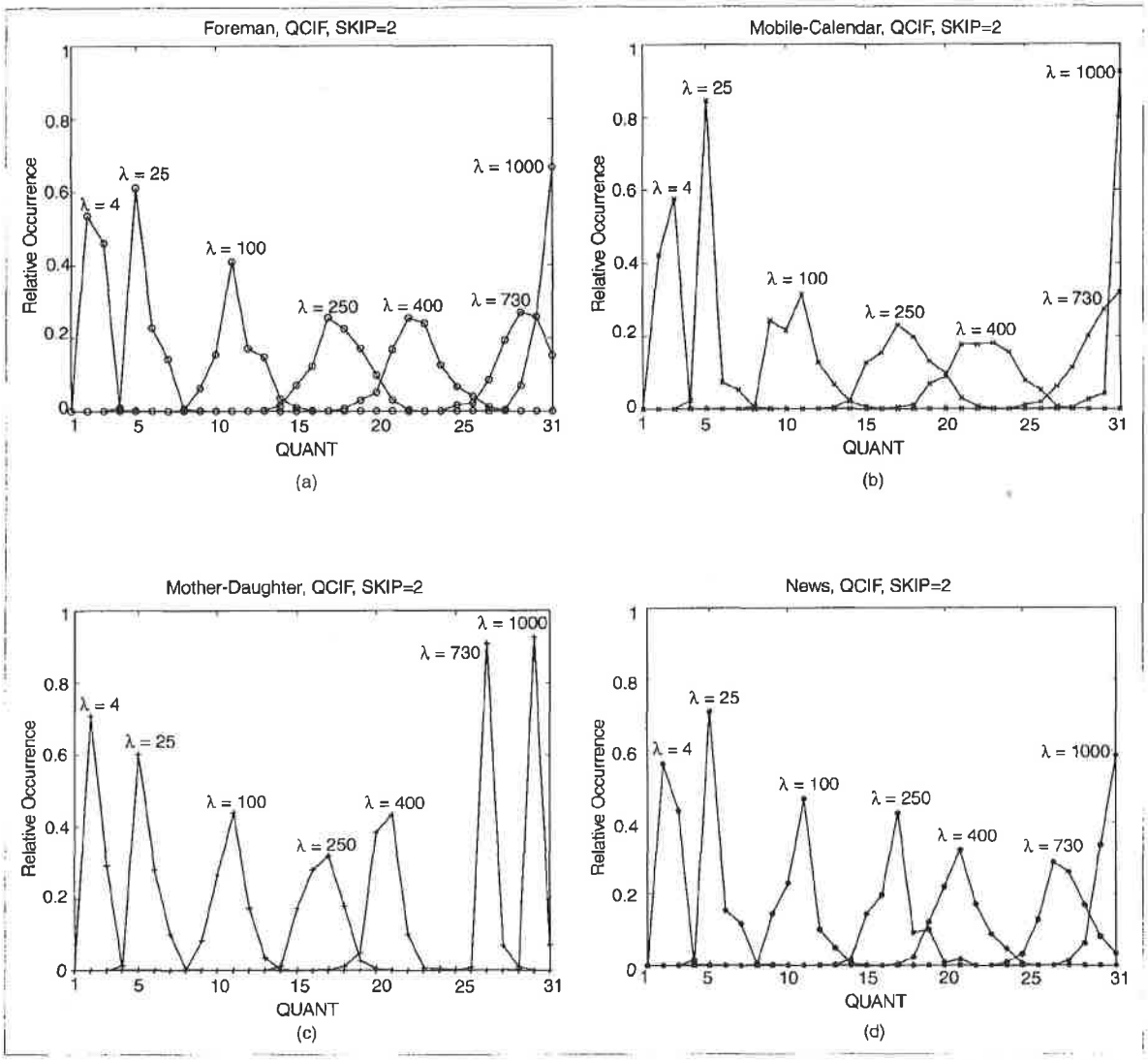
Table 1. Bit-rate partition of the various H.263 modes.

Mode	Motion Coding Bit Rate [%]	Texture Coding Bit Rate [%]
INTRA	0	100
SKIP	100	0
INTER	30 ± 15	70 ∓ 15
INTER+4V	50 ± 20	50 ∓ 20

value on a macroblock-to-macroblock basis appeared by Schuster and Katsaggelos [54].

Quantization

After DCT transformation, the residual signal must be quantized to form the final estimate. Ideally, the choice of quantizer step size Q should be optimized in a rate-distortion sense. Given a quantizer step size Q, the quantization of the residual signal (the mapping of the transformed samples to quantization index values) should also be rate-distortion optimized. The choice of the quantizer output level sent for a given input value should balance the needs of rate and distortion. A simple way to do this is to move the decision thresholds of the quantizer somewhat toward lower bit-rate indices [24, 25, 55]. This is the method used in the ITU-T test model [33]. Alternatively, a $D + \lambda R$ decision can be made explicitly to choose the quantization index. However, in modern video coders such as H.263 the bit rate needed to represent a given quantization index depends not only on the index chosen for a particular sample, but on the values of neighboring quantized indices as well (due to the structure of the coefficient index entropy coding method used). The best performance can be obtained by accounting for these interactions [29]. In recent video coder designs, the interactions have become complex, such that a trellis-based quantization technique may be justified. Such a quantization scheme was proposed by Ortega and Ramchandran [56], and a version that handles the more



▲ *5. Relative occurrence vs. macroblock QUANT for various Lagrange parameter settings. The relative occurrences of macroblock QUANT values are gathered while coding 100 frames of the video sequences Foreman (a), Mobile & Calendar (b), Mother & Daughter (c), and News (d).*

complex structure of the entropy coding of H.263 has recently appeared [57, 58]. Trellis-based quantization was reported to provide approximately a 3% reduction in the bit rate needed for a given level of fidelity when applied to H.263-based DCT coding [57, 58].

Choosing λ and the Quantization Step Size Q

The algorithm for the rate-constrained mode decision can be modified in order to incorporate macroblock quantization step-size changes. For that, the set of macroblock modes to choose from can be extended by also including the prediction mode type INTER+Q for each macroblock, which permits changing Q by a small amount when sending an INTER macroblock. More precisely, for each macroblock a mode M can be chosen from the set

$$M \in \{\text{INTRA, SKIP, INTER, INTER+4V, ...INTER+Q(-4)}, \text{INTER+Q(-2), INTER+Q(+2), INTER+Q(+4)}\} \quad (12)$$

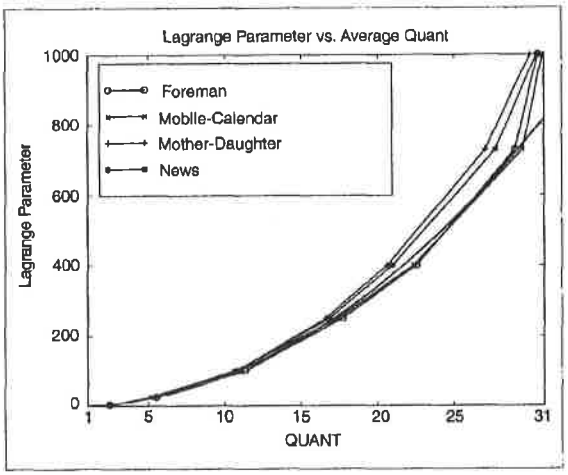
where, for example, INTER+Q(-2) stands for the INTER mode being coded with quantizer step size reduced by two relative to the previous macroblock. Hence, the macroblock Q selected by the minimization routine becomes dependent on λ_{MODE}. Otherwise the algorithm for running the rate-distortion optimized coder remains unchanged.

Figure 5 shows the relative occurrence of macroblock QUANT values (as QUANT is defined in H.263, Q is 2 QUANT) for several Lagrange parameter settings. The Lagrange parameter λ_{MODE} is varied over seven values: 4, 25, 100, 250, 400, 730, 1000, producing seven normalized histograms that are depicted in the plots in Fig. 5. In Fig. 5, the macroblock QUANT values are gathered while coding 100 frames of the video sequences *Foreman*, *Mobile & Calendar*, *Mother & Daughter*, and *News*.

Figure 6 shows the obtained average macroblock QUANT gathered when coding the complete sequences *Foreman*, *Mobile & Calendar*, *Mother & Daughter*, and *News*. The red curve relates to the function

$$\lambda_{MODE} = 0.85 \cdot (\text{QUANT})^2, \quad (13)$$

which is an approximation of the functional relationship between the macroblock QUANT and the Lagrange parameter λ_{MODE} up to QUANT values of 25, and H.263 allows only a choice of QUANT $\in \{1,2,\ldots,31\}$. Particularly remarkable is the strong dependency between λ_{MODE} and QUANT, even for sequences with widely varying content. Note, however, that for a given value of λ_{MODE}, the chosen QUANT tends to be higher for sequences that require higher amounts of bits (*Mobile & Calendar*) in comparison to sequences requiring smaller amounts of bits for coding at that particular λ_{MODE} (*Mother & Daughter*)—but these differences are rather small.



▲ 6. Language parameter λ_{MODE} vs. average macroblock QUANT.

As a further justification of our simple approximation of the relationship between λ_{MODE} and Q, let us assume a typical quantization curve high-rate approximation [59, 60] as follows

$$R(D) = a \ln\left(\frac{\sigma^2}{D}\right), \quad (14)$$

where a is a constant that depends on the source pdf. The minimization (Eq. (8)) for a given value of λ_{MODE} can then be accomplished by setting the derivative of J with respect to D equal to zero. This is equivalent to setting the derivative of $R(D)$ with respect to D equal to $-1 / \lambda_{MODE}$, which yields

$$\frac{dR(D)}{dD} = -\frac{a}{D} \triangleq -\frac{1}{\lambda_{MODE}}. \quad (15)$$

At sufficiently high rates, a reasonably well-behaved source probability distribution can be approximated as a constant within each quantization interval [60]. This leads readily to the typical high bit-rate approximation $D \cong (2 \cdot \text{QUANT})^2 / 12$. The approximations then yield

$$\lambda_{MODE} \cong c \cdot (\text{QUANT})^2, \quad (16)$$

where $c = 4 / (12a)$. Although our assumptions may not be completely realistic, the derivation reveals at least the qualitative insight that it may be reasonable for the value of the Lagrange parameter λ_{MODE} to be proportional to the square of the quantization parameter. As shown above, 0.85 appears to be a reasonable value for use as the constant c.

This ties together two of the three optimization parameters, QUANT and λ_{MODE}. For the third, λ_{MOTION}, we make an adjustment to the relationship to allow use of the SAD measure rather than the SSD measure in that stage of encoding. Experimentally, we have found that an effec-

tive such method is to measure distortion during motion estimation using SAD and to simply adjust λ for the lack of the squaring operation in the error computation, as given by

$$\lambda_{\text{MOTION}} = \sqrt{\lambda_{\text{MODE}}}. \tag{17}$$

This strong dependency that we have thus derived between QUANT, λ_{MODE}, and λ_{MOTION} offers a simple treatment of each of these quantities as a dependent variable of another. For example, the rate control method may adjust the macroblock QUANT occasionally so as to control the average bit rate of a video sequence, while treating λ_{MODE} and λ_{MOTION} as dependent variables using Eqs. (13) and (17). In the experiments reported herein, we therefore used the approximation (17) with the SAD error measure for motion estimation and the approximation (13) with the SSD error measure for mode decisions.

Comparison to Other Encoding Strategies

The ITU-T Video Coding Experts Group (ITU-T Q.15/SG16) maintains an internal document describing examples of encoding strategies, which is called its test model [33, 61]. The mode decision and motion estimation optimization strategies described above, along with the method of choosing λ_{MODE} based on quantizer step size as shown above were recently proposed by the second author and others for inclusion into this test model [62, 63]. The group, which is chaired by the first author, had previously been using a less-optimized encoding approach for its internal evaluations [61], but accepted these methods in the creation of a more recent model [33]. The test model documents, the other referenced Q.15 documents, and other information relating to ITU-T Video Coding Experts Group work can be found on an ftp site maintained by the group (ftp://standard.pictel.com/video-site). Reference software for the test model is available by ftp from the University of British Columbia (ftp://dspftp.ee.ubc.ca/pub/tmn, with further information at http://www.ece.ubc.ca/spmg/research/motion/h263plus).

The less-sophisticated TMN-9 mode-decision method is based on thresholds. It compared the sum of absolute differences of the 16×16 macroblock (W) with respect to its mean value to the minimum prediction SAD obtained by an integer-pixel motion search in order to make its decision between INTRA and INTER modes according to whether

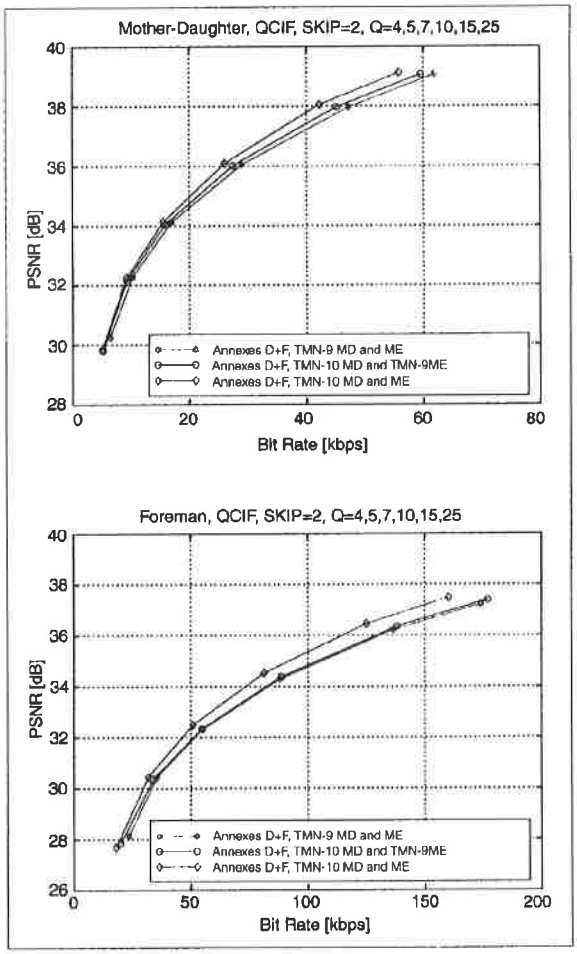
$$W < \min\{\text{SAD(full-pixel}, 16 \times 16)\} - 500. \tag{18}$$

When this inequality was satisfied, the INTRA mode would be chosen for that particular macroblock. The $\min\{\text{SAD(full-pixel}, 16 \times 16)\}$ value above corresponds to the minimum SAD value after integer-pixel motion compensation using a 16×16 motion compensation block size, where the SAD value of the (0,0) MV is reduced by 100 to bias the decision toward choosing the SKIP mode. If the INTER mode is chosen (i.e., if the inequality above is not satisfied) the chosen integer-pixel MV is half-pixel refined. The MVs for INTER+4V blocks were found by half-pixel refining the integer-pixel MV of the 16×16 blocks. Finally, the INTER+4V mode was chosen if

$$\sum_{i=0}^{3} \min\{\text{SAD}_i(\text{half-pixel}, 8 \times 8)\} < \min\{\text{SAD(half-pixel}, 16 \times 16)\} - 200. \tag{19}$$

was satisfied, where $\min\{\text{SAD}_i(\text{half-pixel}, 8 \times 8)\}$ is the minimum SAD value of the ith of the four 8×8 blocks. The SKIP mode was chosen in TMN-9 only if the INTER mode was chosen as better than the INTRA mode and the MV components and all of the quantized transform coefficients were zero.

In the TMN-10 rate-distortion optimized strategy, the motion search uses rate-constrained motion estimation for first finding the best integer-pixel MV in the search



▲ *7. Coding performance for the sequences Mother & Daughter (top) and Foreman (bottom) when comparing the TMN-9 to the TMN-10 encoding strategy*

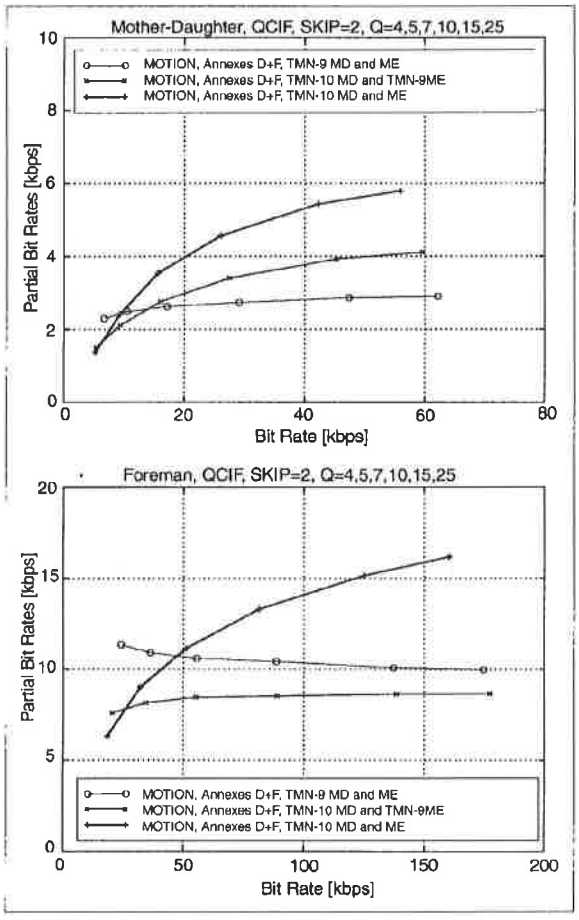
range of ±15 pixels. Then, the best integer-pixel MV is half-pixel refined by again minimizing the Lagrangian cost functional for motion estimation given in Eq. (9). This procedure is executed for both 16×16 and 8×8 blocks. The mode decision of TMN-10 is then conducted using the rate-distortion optimized method described in this article.

The role of the encoding strategy is demonstrated in Fig. 7 for the video sequences *Mother & Daughter* and *Foreman*. The same syntax (H.263 using annexes D and F) was used throughout, with changes only in the mode decision and motion estimation coding methods. These changes are:

▲ Case 1: TMN-9 mode decision and TMN-9 motion estimation

▲ Case 2: TMN-10 mode decision and TMN-9 motion estimation

▲ Case 3: TMN-10 mode decision and TMN-10 motion estimation



▲ *8. Bit-rate partition of motion vectors vs. bit rate for the sequences Mother & Daughter (top) and Foreman (bottom) when employing TMN-10 mode decision and motion estimation.*

Case 2 has been included to demonstrate the impact of rate-constrained mode decision and motion estimation separately. Comparing the three cases, we find that the usage of the full-motion estimation search range of ±15 pixels for the 8×8 block displacement vectors in INTER+4V mode provides most of the gain for the TMN-10 encoding strategy. The INTER+4V prediction mode is very seldom used in TMN-9, indicating that the TMN-9 motion estimation and mode decision rules basically fail to make effective use of this mode. in the highly active *Foreman* sequence, TMN-10 (Case 3) uses this mode for about 15% of macroblocks, whereas TMN-9 (Case 1) uses it for only about 2%.

The TMN-9 motion estimation strategy only permits the use of half-pixel positions for the 8×8 block displacement vectors that immediately surround the previously selected 16×16 block displacement vector that is searched in a ±15 range. We have observed that using the full search range for the 8×8 block displacement vectors leads to improved coding performance for the rate-constrained motion estimation, whereas for the TMN-9 motion estimation, using the full search for this small block size would actually harm the TMN-9 results, since no rate constraint was employed in its search. Only adding a rate constraint to the motion estimation can allow the INTER+4V mode to perform with its full potential.

Figure 8 shows that the TMN-10 coder uses about twice as many bits for motion than the other two coders in order to obtain a better prediction so it can use less difference coding and still obtain an improvement in the overall performance. This is partly because of more frequent use of the INTER+4V mode and partly because of the larger motion estimation search range considered for the 8×8 blocks when the INTER+4V mode is chosen.

In the TMN-10 strategy, the bit rate allocated to the motion part of the information increases as the overall bit rate increases, which makes intuitive sense. The TMN-9 motion estimation shows completely different and sometimes counterintuitive behavior. For the sequence *Foreman*, the MV bit rate actually decreases as overall bit rate increases. This results from the fact that the TMN-9 motion estimation does not employ a rate constraint and that motion estimation is performed using the reconstructed frames (for TMN-9 as well as for TMN-10). As bit rate decreases, these reconstructed frames get noisier, and, since the regularization by the rate constraint is missing for the TMN-9 motion estimation, the estimates for the motion data get noisier and require a higher bit rate.

Rate-constrained mode decision, as employed in TMN-10, provides rather minor gains, but is conceptually simple and introduces a reasonably small computational overhead for some implementations. The overall performance gain of the improved mode decision and motion-estimation methods is typically around 10% in bit rate, or 0.5 dB in PSNR.

Conclusions

We have described the structure of typical video coders and showed that their design and operation requires a keen understanding and analysis of the trade-offs between bit rate and distortion. The single powerful principle of $D + \lambda R$ Lagrange multiplier optimization [22] has emerged as the weapon of choice in the optimization of such systems and can provide significant benefits if judiciously applied.

Acknowledgments

The authors wish to thank Klaus Stuhlmüller, Niko Färber, Bernd Girod, Barry Andrews, Philip Chou, and the PictureTel research team for their support and useful discussions. They also wish to thank guest editors Antonio Ortega and Kannan Ramchandran, as well as Jonathan Su, John Villasenor and his UCLA research team, Faouzi Kossentini and his UBC research team including Michael Gallant, and the anonymous reviewers for their valuable comments. The work of Thomas Wiegand was partially funded by 8×8, Inc.

Gary Sullivan is the manager of communication core research with PictureTel Corporation in Andover, Massachussetts, USA. *Thomas Weigand* is a Ph.D. student with the University of Erlangen-Nuremberg in Erlangen, Germany.

References

1. ITU-T (formerly CCITT) and ISO/IEC JTC1, "Digital Compression and Coding of Continuous-Tone Still Images," ISO/IEC 10918-1 - ITU-T Recommendation T.81 (JPEG), Sept. 1992.
2. W.B. Pennebaker and J.L. Mitchell, *JPEG: Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, USA, 1993.
3. ITU-T (formerly CCITT), "Codec for Videoconferencing Using Primary Digital Group Transmission," ITU-T Recommendation H.120; version 1, 1984; version 2, 1988.
4. ITU-T (formerly CCITT), "Video codec for audiovisual services at $p \times 64$ kbit/s," ITU-T Recommendation H.261; version 1, Nov. 1990; version 2, Mar. 1993.
5. M.L. Liou, "Overview of the $p \times 64$ kbps video coding standard," *Communications of the ACM*, vol. 34, pp.47-58, Apr.1991.
6. ISO/IEC JTC1, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 2: Video," ISO/IEC 11172-2 (MPEG-1), Mar. 1993.
7. J.L. Mitchell, W.B. Pennebaker, C. Fogg, and D.J. LeGall, *MPEG Video Compression Standard*, Chapman and Hall, New York, USA, 1997.
8. ITU-T (formerly CCITT) and ISO/IEC JTC1, "Generic coding of moving pictures and associated audio information - Part 2: Video," ITU-T Recommendation H.262 - ISO/IEC 13818-2 (MPEG-2), Nov. 1994.
9. B.G. Haskell, A. Puri, A.N. Netravalli, *Digital Video: An Introduction to MPEG-2*, Chapman and Hall, New York, USA, 1997.
10. ITU-T (formerly CCITT), "Video coding for low bitrate communication," ITU-T Recommendation H.263; version 1, Nov. 1995; version 2, Jan. 1998.
11. K. Rijkse, "H.263: video coding for low-bit-rate communication," *IEEE Commun. Mag.*, vol. 34, no. 12, pp. 42-45, Dec. 1996.
12. B. Girod, E. Steinbach, and N. Färber, "Performance of the H.263 video compression standard," *Journal of VLSI Signal Processing: Systems for Signal, Image, and Video Technology*, 1997.
13. B. Erol, M. Gallant, G. Côté, and F. Kossentini, "The H.263+ video coding standard: complexity and performance", in *Proceedings of the IEEE Data Compression Conference*, Snowbird, Utah, USA, pp. 259-268, Mar. 1998.
14. A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video Compression: An Overview," this issue, pp. 23-50.
15. G.M. Schuster, G. Melnikov, and A.K. Katsaggelos, "Operationally vertex-based shape coding," this issue, pp. 91-108.
16. H. Watanabe and S. Singhal, "Windowed motion compensation," in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, vol. 1605, pp. 582-589, 1991.
17. S. Nogaki and M. Ohta, "An overlapped block motion compensation for high quality motion picture coding", in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 1, pp. 184-187, May 1992.
18. G.J. Sullivan, "Multi-hypothesis motion compensation for low bit-rate video coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, MN, USA, vol. 5, pp. 437-440, April, 1993.
19. M.T. Orchard and G.J. Sullivan, "Overlapped block motion compensation: an estimation-theoretic approach," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 693-699, Sept. 1994.
20. K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders", *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 533-545, Sept. 1994.
21. J. Lee and B.W. Dickinson, "Joint optimization of frame type selection and bit allocation for MPEG video coders," in *Proceedings of the IEEE International Conference on Image Processing*, Austin, USA, vol. 2, pp. 962-966, Nov. 1994.
22. Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, pp. 1445-1453, Sept. 1988.
23. H. Everett III, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, pp. 399-417, 1963.
24. P.A. Chou, T. Lookabaugh, and R.M. Gray, "Entropy-constrained vector quantization," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 1, pp. 31-42, Jan. 1989.
25. A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, USA, 1991.
26. T. Watanabe, Y. Tsukuhara, and K. Ohzeki, "Rate-adaptive DCT coding for color picture," in *Proceedings of the Picture Coding Symposium*, Boston, MA, paper no. 3.13, Mar. 1990.
27. S.-W. Wu and A. Gersho, "Rate-constrained optimal block-adaptive coding for digital tape recording of HDTV", *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 100-112, vol. 1, no. 1, Mar. 1991
28. S.-W. Wu and A. Gersho, "Enhanced video compression with standardized bit stream syntax," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Minneapolis, MN, USA, vol. 1, pp. 103-106, Apr. 1993.
29. S.-W. Wu, *Enhanced Image and Video Compression with Constraints on the Bit Stream Format*, Ph.D. thesis, U. C., Santa Barbara, Mar. 1993.
30. G.J. Sullivan and R.L. Baker, "Efficient quadtree coding of images and video," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, Canada, pp. 2661-2664, May 1991.

31. G.J. Sullivan and R.L. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," in *Global Telecomm. Conf. (GLOBECOM'91)*, pp. 85-90, Dec. 1991.

32. B. Girod, "Rate-constrained motion estimation," in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, Chicago, USA, pp. 1026-1034, Sept. 1994.

33. ITU-T SG16/Q15 (T. Gardos, ed.), "Video codec test model number 10 (TMN-10)," ITU-T SG16/Q15 document Q15-D-65, (downloadable via ftp://standard.pictel.com/video-site), Apr. 1998.

34. J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled lagrange multiplier method," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 546-558, Sept. 1994.

35. A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Transactions on Image Processing*, vol. 3, no. 1, pp. 26-40, Jan. 1994.

36. J. Ribas-Corbera and S. Lei, "Rate control for low-delay video communications," ITU-T SG16/Q15 document Q15-A-20, (downloadable via ftp://standard.pictel.com/video-site), June 1997.

37. M.C. Chen and A.N. Willson, "Rate-distortion optimal motion estimation for video coding," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, USA, vol. 4, pp. 2096-2099, May 1996.

38. M.C. Chen and A.N. Willson, "Design and optimization of a differentially coded variable block size motion compensation system," *Proceedings of the IEEE International Conference on Image Processing*, Laussanne, Switzerland, vol. 3, pp. 259-262, Sept. 1996.

39. M.C. Chen and A.N. Willson, "Rate-distortion optimal motion estimation algorithms for motion-compensated transform video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 2, pp. 147-158, Apr. 1998.

40. W.C. Chung, F. Kossentini, and M.J.T. Smith, "An efficient motion estimation technique based on a rate-distortion criterion," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Atlanta, USA, vol. 4, pp. 1926-1929, May 1996.

41. F. Kossentini, Y.-W. Lee, M.J.T. Smith, and R. Ward, "Predictive RD optimized motion estimation for very low bit rate video coding," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 9, pp. 1752-1763, Dec. 1997.

42. G.M. Schuster and A.K. Katsaggelos, "A video compression scheme with optimal bit allocation among segmentation, motion, and residual error," *IEEE Transactions on Image Processing*, vol. 6, pp. 1487-1502, Nov. 1997.

43. B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences", *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 7, pp. 1140-1154, Aug. 1987.

44. B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Transactions on Communications*, vol. 41, no. 4, pp. 604-612, Apr. 1993.

45. H.G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding," *Proceedings of the IEEE*, vol. 73, no. 9, pp. 523-548, Apr. 1985.

46. B. Girod, "Efficiency analysis of multi-hypothesis motion-compensated prediction for video coding," *IEEE Transactions on Image Processing*, 1997, submitted for publication.

47. M. Flierl, T. Wiegand, and B. Girod, "A local optimal design algorithm for block-based multi-hypothesis motion-compensated prediction," in *Proc. DCC*, Snowbird, USA, Mar. 1998.

48. T. Wiegand, M. Flierl, and B. Girod, "Entropy-constrained linear vector prediction for motion-compensated video coding", in *Proc. ISIT*, Boston, USA, Aug. 1998.

49. T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, Sept. 1998.

50. Nokia Research Center (P. Haavisto, et al.), "Proposal for efficient coding," ISO/IEC JTC1/SC29/WG11, MPEG document MPEG96/M0904 , July 1996.

51. M. Karczewicz, J. Nieweglowski, and P. Haavisto, "Video coding using motion compensation with polynomial motion vector fields", *Signal Processing: Image Communication*, vol. 10, pp. 63-91, 1997.

52. T. Wiegand, M. Lightstone, T.G. Campbell, and S.K. Mitra, "Efficient mode selection for block-based motion compensated video coding", in *Proceedings of the IEEE International Conference on Image Processing*, Washington, D.C., USA, Oct. 1995.

53. T. Wiegand, M. Lightstone, D. Mukherjee, T.G. Campbell, and S.K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 182-190, Apr. 1996.

54. G.M. Schuster and A.K. Katsaggelos, "Fast and efficient mode and quantizer selection in the rate distortion sense for H.263," in *Proceedings of the SPIE Conference on Visual Communications and Image Processing*, Orlando, USA, pp. 784-795, Mar. 1996.

55. G.J. Sullivan, "Efficient scalar quantization of exponential and Laplacian random variables," *IEEE Transactions on Information Theory*, vol. 42, no. 5, pp. 1365-1374, Sept. 1996.

56. A. Ortega and K. Ramchandran, "Forward-adaptive quantization with optimal overhead cost for image and video coding with applications to mpeg video coders," in *Proceedings of the SPIE, Digital Video Compression: Algorithms and Technologies*, San Jose, USA, Feb. 1995.

57. J. Wen, M. Luttrell, and J. Villasenor, "Simulation results on trellis-based adaptive quantization," ITU-T SG16/Q15 document Q15-D-40, (downloadable via ftp://standard.pictel.com/video-site), Apr. 1998.

58. J. Wen, M. Luttrell, and J.Villasenor, "Trellis-Based R-D Optimal Quantization in H.263+," *IEEE Transactions on Image Processing*, 1998, submitted for publication.

59. N.S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, Englewood Cliffs, USA, 1984.

60. H. Gish and J.N. Pierce, "Asymptotically efficient quantizing", *IEEE Transactions on Information Theory*, vol. 14, pp. 676-683, Sept. 1968.

61. ITU-T SG16/Q15 (T. Gardos, ed.), "Video codec test model number 9 (TMN-9)," ITU-T SG16/Q15 document Q15-C-15, (downloadable via ftp://standard.pictel.com/video-site), Dec. 1997.

62. T. Wiegand and B.D. Andrews, "An improved H.263 coder using rate-distortion optimization," ITU-T SG16/Q15 document Q15-D-13, (downloadable via ftp://standard.pictel.com/video-site), Apr. 1998.

63. M. Gallant, G. Coté, and F. Kossentini, "Description of and results for rate-distortion-based coder," ITU-T SG16/Q15 document Q15-D-49, (downloadable via ftp://standard.pictel.com/video-site), Apr. 1998.

Exhibit AA

To

Joint Claim Chart

Removal of Motion Uncertainty and Quantization Noise in Motion Compensation

Bo Tao, *Member, IEEE,* and Michael T. Orchard, *Fellow, IEEE*

***Abstract*—In a standard hybrid video coder, there are two important factors affecting motion-compensated prediction: motion uncertainty and quantization noise. Motion uncertainty results from the use of block-decimated motion compensation, which cannot specify the motion for each pixel. The propagation of quantization noise results from interframe prediction. We analyze both these effects, and their block-structured nonstationary properties. We propose a block-adaptive linear filtering framework to reduce motion uncertainty and quantization noise propagation simultaneously. This new motion-compensated predictor can be viewed as joint application of overlapped block motion compensation and loop filtering (LF). Several system configurations are evaluated. We show that this linear filtering framework achieves better rate-distortion performance than the single use of either overlapped block motion compensation or LF.**

***Index Terms*—Loop filtering, motion compensation, motion uncertainty, overlapped block motion compensation, quantization noise.**

I. Introduction

BLOCK motion compensation is widely used in video encoders, and is incorporated in many industry standards, such as ISO MPEG-1, 2, ITU-T H.261, and H.263. In a traditional block motion-compensated encoder, each block in a picture is assigned a motion estimate obtained using some motion estimation algorithm, and the prediction for this block is given by the block in the previous decoded picture pointed to by this estimated motion.

Typically, the actual motion varies within a block, and hence the block motion estimate cannot specify the correct motion for all pixels in a block. Therefore, not all pixels have accurate motion-compensated prediction. We call this *motion uncertainty*. This can be viewed as adding noise to motion compensation. Another noise source affecting motion compensation is *quantization noise*, which refers to the error introduced when transform coefficients are quantized. With quantization noise present in the previous reconstructed picture, even if the motion is correct at every pixel, motion compensation predicts each block with the decoded previous picture rather than the actual previous picture. Thus, quantization error from the previous decoding propagates to the current predicted picture.

Manuscript received February 19, 1999; revised April 12, 2000. This paper was recommended by Associate Editor H. Watanabe.

B. Tao was with Electrical Engineering Department, Princeton University, Princeton, NJ 08544 USA. He is now with Streaming21 Inc., Los Gatos, CA 95124 USA (e-mail: Bo.Tao@streaming21.com).

M. T. Orchard is with Electrical Engineering Department, Princeton University, Princeton, NJ 08544 USA (e-mail: orchard@ee.princeton.edu).

Publisher Item Identifier S 1051-8215(01)00672-3.

Both motion uncertainty and quantization noise reduce the accuracy of the motion-compensated prediction. Linear filters can be used to reduce these noise sources, and hence to improve the motion compensation performance. Girod [1] studies a framework for filtering these noise sources by applying a filter within the encoding loop [denoted *loop filtering* (LF)]. Though [1] optimizes the filter with respect to the noise models (of both motion estimation error and quantization noise), this Wiener-type linear filtering approach is based on an underlying stationarity assumption that limits its effectiveness. Similar can be said to the work in [2]. Non-adaptive linear filtering applied at all pixel locations in the predicted frame is best suited for removing noise processes whose statistics do not depend on pixel location. Rather, in this work, we will model the block nonstationary structure of noise contributed by motion uncertainty and quantization noise, and to propose a block-adaptive linear filtering framework for improving the accuracy of motion-compensated prediction.

Overlapped block motion compensation (OBMC) has been proposed in the literature to improve the performance of motion compensation [3], [4], and various forms of OBMC have been incorporated into several recent video coding standards [5]. Orchard and Sullivan [6] recognized and exploited the block nonstationary structure of motion uncertainty, and proposed a framework to optimize OBMC windows. Experimental results in [6] graphically depict the block structure of error statistics caused by motion uncertainty, and the window design algorithm it proposes is based on optimizing a collection of component filters, one for each pixel position within the block. Thus, the optimization of each component filter reflects the local statistics of motion uncertainty noise at each pixel position.

While the approach of [6] is well suited for exploiting the nonstationarity of motion uncertainty, it does not provide a good filtering framework for reducing quantization noise. This is because, although quantization noise also generates block-structured noise statistics, it does so with respect to blocks that are, in general, not aligned with the block structure of the motion-compensated prediction. Quantization noise in the hybrid video coder is generated by the transform coding of residual signal, appearing as block-structured error in the previous decoded frame. Thus, quantization noise generates local statistics that depend on a pixel's position relative to the previous decoded frame's block grid. In moving pixel values to random positions in the predicted frame, motion compensation loses track of the nonstationary structure produced by quantization noise.

This paper extends our previous work in [7], where the idea of joint application of space-varying OBMC and LF was first introduced. In this paper, we will present a more rigorous model

for motion uncertainty. Quantization noise in a hybrid video coder is examined more thoroughly. We will also discuss how the OBMC filters (or equivalently, the overlapped window) and loop filters change when they are used simultaneously. The changes in the shape of these filters give us more insight into the interaction of motion uncertainty and quantization noise in the design of linear filters in a motion-compensated predictor.

Section II provides a mathematical model for motion uncertainty, which leads to the discussion of the associated block-structured noise statistics. This model is established by analyzing the relationship between the block motion estimate and the motion random field. The next section analyzes the nonstationary block structure of quantization noise, especially in the video coding environments. These analyzes motivate us to propose a block-adaptive linear filtering framework, which can be viewed as jointly applying OBMC and LF. Several approaches to joint OBMC and LF in a video encoder are evaluated to exploit the nonstationary noise statistics of motion uncertainty and quantization noise in Section IV. Section V shows that better rate-distortion performance is achieved with such a joint application. In this section, we also discuss the interaction of motion uncertainty and quantization noise by examining the filter changes when OBMC and LF are applied simultaneously. The discussion in the last section concludes this paper.

II. Motion Uncertainty

Motion uncertainty reflects the fact that, when pixels in a block undergo different motion, block motion estimation cannot resolve the uncertainty associated with the motion at each pixel. This section characterizes the nonstationary block-structured statistics resulting from motion uncertainty. We begin by studying the block motion estimate first.

Let us start with one-dimensional signal $\{y_k(i)\}$ at time k. The motion of pixel i is $v_i, y_k(i) = y_{k-1}(i+v_i)$. Let the derivative at pixel $i+v_i$ in frame $k-1$ be d_i. For a block of pixels, motion estimation with minimum Euclidean distance block matching gives

$$\begin{aligned}\hat{v} &= \arg\min_v \sum_{i\subset B}[y_k(i) - y_{k-1}(i+v)]^2 \\ &= \arg\min_v \sum_{i\subset B}[y_{k-1}(i+v_i) - y_{k-1}(i+v)]^2 \\ &\approx \arg\min_v \sum_{i\subset B}[y_{k-1}(i+v_i) - (y_{k-1}(i+v_i) \\ &\quad + d_i(v-v_i))]^2 \\ &= \arg\min_v \sum_{i\subset B}[d_i(v-v_i)]^2 \\ &= \frac{\sum_{i\subset B} d_i^2 \cdot v_i}{\sum_{i\in B} d_i^2} \end{aligned} \tag{1}$$

where B is the support of a block, and a Taylor's series expansion is used for approximation.

Block matching with l_2 distance minimizes the energy of the residual signal, and for Gaussian signals, minimizes the bit-rate required to encode the residual signal. In practice, block matching with l_1 distance is often used for fast computation reasons. It can be viewed as an approximation to block matching with l_2 distance with the l_1 distance being bounded by the l_2 distance through Cauchy–Schwartz inequality. Furthermore, the minimization process implicitly assumes that full search is used in motion estimation. Practical fast search algorithms can be thought of as a fast approximate solution to the minimization problem. Thus, the equations above apply to real coders.

If we can assume that the scene intensity field is such that d_i^2 are identically distributed, then

$$E_I\left(\frac{d_i^2}{\sum_{i\subset B} d_i^2}\right) = \frac{1}{|B|}.$$

If we further assume the intensity field $\{y_k(i)\}$ and the motion field $\{v_i\}$ are independent of each other, we have

$$E_I[\hat{v} \mid \{v_i\}_{i\subset B}] = \frac{1}{|B|}\sum_{i\subset B} v_i \tag{2}$$

up to the accuracy of the first-order Taylor's expansion, where the expectation E_I is taken with respect to the distribution of the intensity field.

The above result can be easily generalized to two-dimensional signals by using the gradient at each pixel. Let $\{I_k(s), s = (r,c)\}$ be the kth frame. Let $v(s)$ be the motion of pixel s in frame $k, I_k(s) = I_{k-1}(s+v(s))$ and $v(s) = (v_r(s), v_c(s))$. By using Taylor's expansion again, we have

$$\begin{aligned}&[I_k(s) - I_{k-1}(s+v)]^2 \\ &\quad\approx \Bigg[I_{k-1}(s+v_s) - I_{k-1}(s+v_s) \\ &\qquad - \frac{\partial I_{k-1}(s+v_s)}{\partial r}(v_r - v_r(s)) \\ &\qquad - \frac{\partial I_{k-1}(s+v_s)}{\partial c}(v_c - v_c(s))\Bigg]^2 \\ &\quad\approx \left[\frac{\partial I_{k-1}(s+v_s)}{\partial r}(v_r - v_r(s))\right]^2 \\ &\qquad + \left[\frac{\partial I_{k-1}(s+v_s)}{\partial c}(v_c - v_c(s))\right]^2.\end{aligned}$$

Compare this with (1), it is easily seen that for two-dimensional signals, we still have the relationship given in (2).

Having analyzed the field of block motion estimates, let us turn our attention to the underlying motion random field $\{v(s)\}$. If the motion for each pixel $s = (r,c)$ in the block satisfies the affine motion model

$$\begin{pmatrix} r \\ c \end{pmatrix} \mapsto A_{2\times 2} \times \begin{pmatrix} r \\ c \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \tag{3}$$

it is easily seen that

$$\begin{aligned}&E_I[\hat{v} \mid \{v(s)\}_{s\subset B}, v(s) \text{ satisfies (3)}] \\ &\quad = v\left(\frac{1}{|B|}\sum_{s\subset B} s\right) \\ &\quad = \frac{1}{|B|}\sum_{s\subset B} v(s).\end{aligned}$$

Note that this is obtained when pixels undergo affine motion and when translational block motion estimation is used. This is a fairly common situation in practice. Hence, in the rest of this section, we will model the block motion estimate as the motion of the block center, i.e.,

$$\hat{v} \approx v\left(\frac{1}{|B|}\sum_{s\in B} s\right). \tag{4}$$

It should be noted that the above result is very different from stationary assumptions about the motion estimate error in previous literature. They often assume that the motion estimation error for all pixels in a block $\{v_s - \hat{v}\}$ is identically distributed [1], [2]. In contrast, our result suggests that the motion estimation error for different pixels in a block should have different distributions. This nonstationarity of motion uncertainty will cause block-structured nonstationarity in motion compensation performance, which will be examined next.

The motion compensation error for pixel s is given by $I_k(s) - I_{k-1}(s+\hat{v})$. Define the autocorrelation function of the scene intensity field by

$$E_I[I_k(s_1)I_k(s_2)] = \mathrm{corr}_I(\|s_1 - s_2\|)$$

where $\mathrm{corr}_I(x)$ decreases with increasing $|x|$. The residual signal energy can be computed as

$$\begin{aligned} &E_M\{E_I[(I_k(s) - I_{k-1}(s+\hat{v}))^2]\} \\ &\quad = E_M\{E_I[(I_{k-1}(s+v_s) - I_{k-1}(s+\hat{v}))^2]\} \\ &\quad = E_M\left\{E_I\left[I_{k-1}^2(s+v_s) + I_{k-1}^2(s+\hat{v})\right.\right. \\ &\qquad \left.\left. - 2I_{k-1}(s+v_s)\times I_{k-1}(s+\hat{v})\right]\right\} \\ &\quad = 2\,\mathrm{corr}_I(0) - 2E_M[\mathrm{corr}_I(\|v_s - \hat{v}\|)] \end{aligned}$$

which is nonstationary and block-structured according to our previous analysis (4). We call this the *block-structured nonstationarity of motion uncertainty*. Furthermore, if the auto-correlation function $\mathrm{corr}_I(\cdot)$ can be approximated with a quadratic function, it can be easily shown that the motion compensation error is larger for boundary pixels than for pixels around the center of the block [8]. However, we elect not to make such a digression, since our goal here is only to show the block-structured nonstationarity of motion uncertainty.

It should be noted that stationarity is an indispensable component in the analysis. In fact, we have both stationarity and nonstationarity at the same time. Within a block, it is nonstationary. On top of that is the block-periodic stationarity. (In other words, the signal is cyclic stationary.) When either stationarity or nonstationarity is used to describe the properties of motion uncertainty, one aspect tends to be emphasized more than the other aspect. But we have chosen to use "block-structured nonstationarity" to emphasize the nonstationarity within a block, which is more important to this work than block-periodic stationarity.

The result above also implies that pixels symmetric with respect to the block center will have the same motion compensation performance, which we call the *motion compensation symmetry*. Such a four-way symmetry will be exploited in our linear filtering framework. These analyses are well supported by the results in Fig. 1, which show the average motion compensation error energy for different pixels in a 16×16 block, computed from a real video sequence. Full-search translational block motion estimation is used to obtain the results.

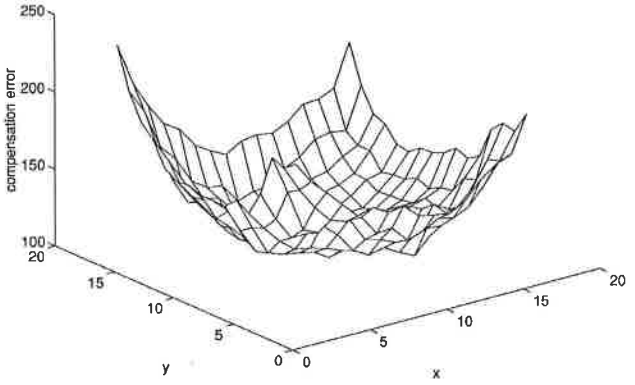


Fig. 1. Energy of motion compensation error in a 16×16 block. The sequence is Football.

III. Quantization Noise

Besides motion uncertainty, quantization noise is another important noise source limiting the compensation performance. This section analyzes the block-nonstationary structure of quantization noise by characterizing the spatial distribution of quantization error energy generated by the coding of the previous frame. Since motion compensation propagates this error into the predicted frame, effective filtering of quantization error improves the efficiency of motion-compensation.

Quantization in discrete cosine transform (DCT) coding of low-frequency image sources produces block-structured error statistics similar to those generated by motion uncertainty. This is due to source statistics and efficient quantization strategy, where most high-frequency DCT coefficients are quantized to zero, leaving the block to be approximated by a few low-frequency coefficients. For slowly varying sources, the low-frequency coefficients provide a best average match for the block center, with error energy increasing with the distance from the center.

As a simple example of analysis, let us consider the situation where only the DC coefficient is used to approximate a block, while all other coefficients are set to 0. For ease of notation, the block of pixels is denoted by $I_k(s)$, $s = (r, c)$, with $-(N/2) \le r, c \le (N/2)$. To approximate a block with the DC coefficient is to approximately represent each $I_k(r,c)$ with $\bar{I}_k = 1/(N+1)^2 \sum_{x,y} I_k(x,y)$. For pixel (r,c) in the block, the expected approximation error is

$$\begin{aligned} &E_I[(I_k(r,c) - \bar{I}_k)^2] \\ &\quad = E_I\left[\left(I_k(r,c) - \frac{1}{(N+1)^2}\sum I_k(x,y)\right)^2\right] \\ &\quad = A - \frac{2}{(N+1)^2}E_I\left[\sum_{x,y} I_k(r,c)I_k(x,y)\right] \\ &\quad = A - \frac{2}{(N+1)^2}\sum_{x,y}\mathrm{corr}_I(\|(r,c) - (x,y)\|) \end{aligned}$$

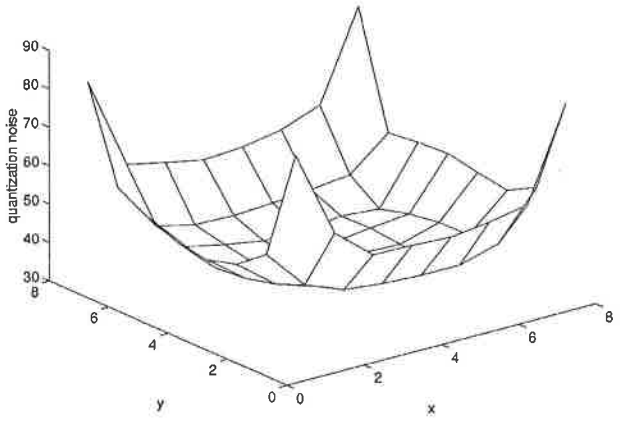


Fig. 2. Energy of pixel-domain quantization noise averaged over Football sequence that is coded using 8 × 8 DCT and scalar quantization.

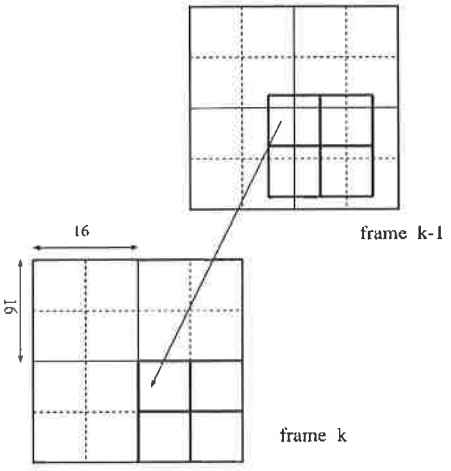


Fig. 3. Block structure of quantization noise in the previous frame does not align with the one in the current frame due to motion compensation.

where $\text{corr}_I(\cdot)$ is the auto-correlation function of the intensity field, and

$$A = E_I\left[I_k^2(r,c)\right] + E_I\left[\left(\frac{1}{(N+1)^2}\sum I_k(x,y)\right)^2\right].$$

A is constant with a stationary assumption on the intensity field. Thus, if $\text{corr}_I(\cdot)$ is monotonically decreasing, as is the situation in practice, then it is easily seen that

$$E_I[(I_k(s_1) - \bar{I}_k)^2] \leq E_I[(I_k(s_2) - \bar{I}_k)^2], \quad \text{if } \|s_1\| \leq \|s_2\|.$$

Fig. 2 gives evidence of this effect, showing the average quantization noise energy as a function of block pixel position for DCT coding of a real video sequence.

Note that these block variations are defined with respect to the block structure of the previous frame. After being propagated to the predicted frame via motion compensation, quantization noise statistics no longer share this structure (Fig. 3). Thus, in the following section, a component of our filters will be adapted to the pixel position in the previous picture to smooth out the noise.

Additional structure in quantization noise statistics can be associated with the interaction of motion-compensated prediction and quantization of the previous frame. In many video standards, motion compensation is applied to 16 × 16 macroblocks, while the residual error is DCT coded with 8×8 blocks. Because of the block-structured errors resulting from motion uncertainty discussed in the previous section, the four 8 × 8 residual blocks within each macroblock do not share the same statistical structure. Rather, rotational symmetry can be identified as analyzed in the previous section. That is, if we order the pixels in each block in such a way that the nth pixels in the four blocks are symmetric to each other with respect to the block center, then the four residual blocks will have the same joint probability distribution.

Let $d(r,c)$ and $n(r,c), 0 \leq r,c \leq 15$ represent the four 8 × 8 blocks of DCT coefficients, and their corresponding DCT-domain quantization noise, respectively. A simple analysis will show that $P[d(r,c) = a] = P[d(r,c-8) = (-1)^c a]$, if $8 \leq c \leq 15$ and $P[d(r,c) = a] = P[d(r-8,c) = (-1)^r a]$, if $8 \leq r \leq 15$, due to motion compensation symmetry, where P represents probability distribution. Therefore, members of the DCT coefficient quadruplet $\{d(r,c), d(r,c-8), d(r-8,c), d(r-8,c-8)\}$, $8 \leq r,c \leq 15$ would have different distributions in general. But in practice, it is most often the case that each AC coefficient has zero mean symmetric distribution. In this case, DCT coefficients in each quadruplet are statistically identical. When the same set of scalar quantizers are used to quantize the four blocks of DCT coefficients as in most industry video coding standards, it is seen that members of the quantization noise quadruplet $\{n(r,c), n(r,c-8), n(r-8,c), n(r-8,c-8)\}$, $8 \leq r,c \leq 15$ are statistically identical, too. Furthermore, if DCT-domain quantization noise are independent from one coefficient position to another, it follows that pixel-domain quantization noise is block-wise periodic in a statistical sense, for 8 × 8 blocks. This periodicity will be exploited in the next section when designing linear filters to reduce the quantization noise.

IV. Block-Adaptive Linear Filtering to Reduce Motion Uncertainty and Quantization Noise

Motivated by the block-structured noise statistics described in the previous two sections, this section presents a unified linear filtering framework aimed at exploiting these nonstationarities to improve prediction accuracy. We take the approach of optimizing a linear prediction of the current frame based on the block motion estimates, the previously decoded frame, and the known nonstationary structure of both the motion uncertainty and the quantization noise.

Following the formulation of OBMC [6], the predicted pixel is computed as a linear combination of a set of candidate motion-based estimates, each generated by a block motion vector from some neighborhood of the pixel location. Because motion uncertainty generates block-structured variations in error energy statistics in the predicted frame, the weights for the linear estimator depend on the pixel position within the block. It has been suggested [6] that improved motion compensation can be achieved by adding spatial neighbors of predicted pixels to form an expanded set of candidate estimates in this formulation, thus gaining access to a type of spatial smoothing. However, the spa-

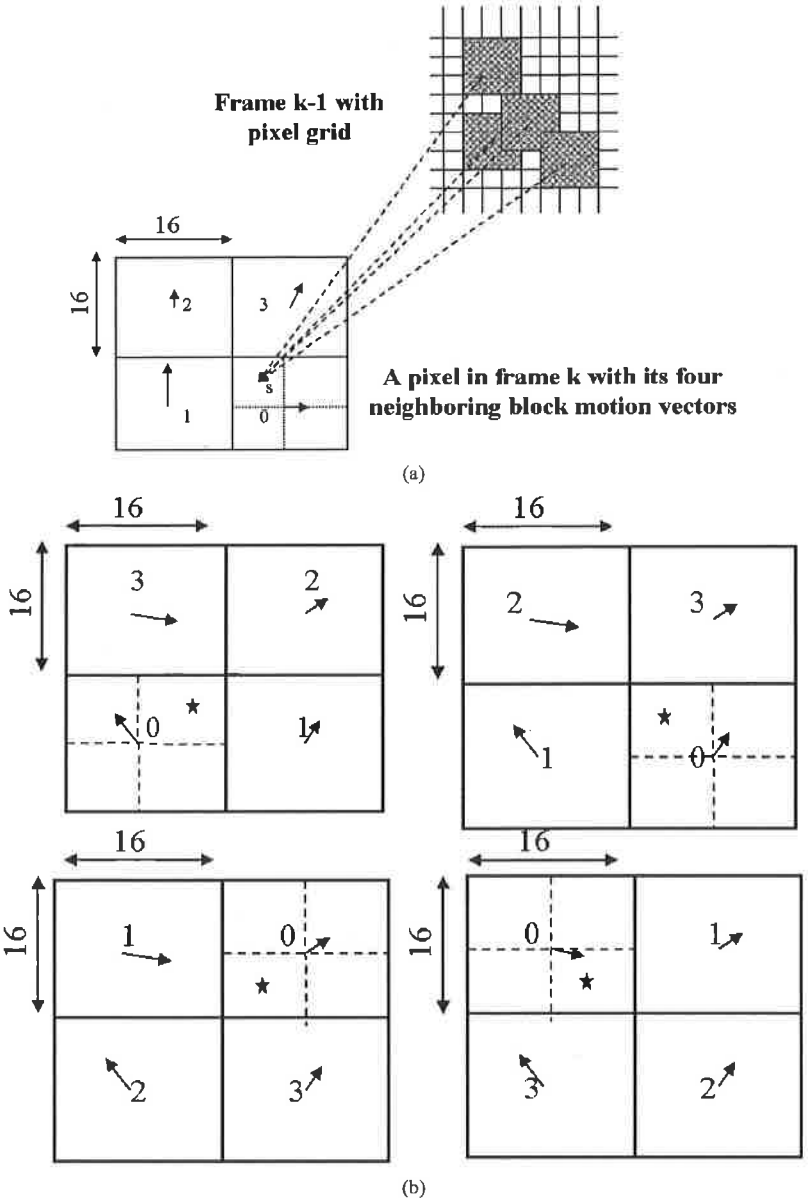


Fig. 4. (a) Motion compensation using OBMC + LF. (b) Indexing of the four neighboring block motion vectors for pixels in each quadrant of the 16 × 16 block. A star indicates the current quadrant.

tial filtering implemented in this way overlaps with the objectives of the spatially varying loop-filtering proposed in the next paragraph. Our experiments found our approach to give better results, and found virtually no gain from using both. Taken alone then, the algorithm used here is identical to OBMC in [6].

To reduce quantization noise in the motion-compensated prediction, we propose optimizing spatial filters applied to the previously decoded frame prior to motion compensation. Because quantization noise generates block-structured variations in error energy concentration, separate filters are optimized for each pixel position in the block structure of the decoded frame. This part of the algorithm can be viewed as spatially-varying LF.

In this work, we use a 32 × 32 OBMC window applied on a 16 × 16 block grid as has been used in previous works. The window is designed as 64 estimators with each estimator being applied at four block positions, exploiting four-way symmetry. This is the same configuration as used in [3] and [6]. The assumption behind this configuration is that, given the motion at the four closest block centers, the motion at a pixel s is independent of the motion at other block centers (see Fig. 4(a)). Therefore, for each pixel, motion compensation is accomplished by using a four-tap estimator that corresponds to a 32 × 32 OBMC window. Furthermore, we use 3 × 3 loop filters, and 64 such filters are designed, one for each 8 × 8 block position. Because of the periodicity of quantization noise discussed in the previous section, the same set of 64 filters will be applied in all 8 × 8 blocks.

Fig. 4(a) illustrates the linear filtering scheme proposed above. Shown in the upper-right corner is the previous frame

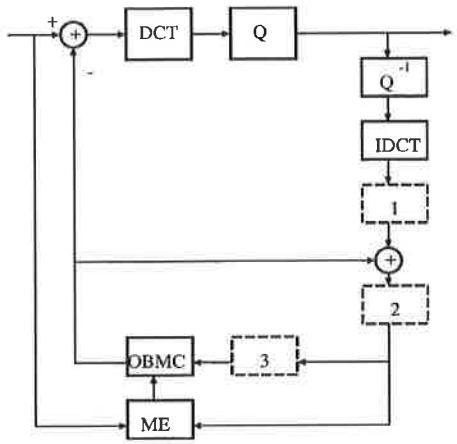


Fig. 5. System configurations. While the placement of OBMC filter is straightforward, three possible positions for placing the loop filter are considered.

in pixel grid. The current frame in block grid is shown in the lower-left corner. Since the OBMC window has size 32×32, the prediction value for each pixel is the weighted sum of four predictions—one for each windowing computation. In other words, each pixel in the current frame has four neighboring block motion vectors. For pixel s, the index of its four neighboring blocks are shown in Fig. 4(a). Each shaded 3×3 area in the figure represents the LF effects, that is, the prediction associated with each block motion vector is itself a weighted average computed using the 3×3 loop filter.

Fig. 5 shows a system diagram of the hybrid video coder. The OBMC filter can only be introduced by replacing the conventional motion compensation block. However, the spatially varying LF can be applied anywhere after the previous frame decoding and before OBMC. Three possible positions for the LF are shown in Fig. 5, and we denote them as the three configurations tested in the following simulations.

Let g_i and $h_i, i = 0, 1, \ldots, 63$, be the ith OBMC filter and loop filter, respectively. Each OBMC filter is represented as an 4×1 vector, and each loop filter as a 9×1 vector. Since signals do not have (local) zero mean in practice, we constrain the filters to satisfy

$$h_i^T u_9 = 1, \quad i = 0, \ldots, 63$$
$$g_i^T u_4 = 1, \quad i = 0, \ldots, 63$$

where u_n is an $n \times 1$ vector with all entries being unit, and T represents the transpose operation. For pixels in the fourth quadrant of the 16×16 block, the filter index of i is illustrated in Fig. 6(a). For pixels in other quadrants, loop filter indexing is the same because of the block periodicity of quantization noise, while OBMC filter indexing is symmetric with respect to the block center because of the symmetry of motion uncertainty. For a 3×3 loop filter, the filter coefficients are indexed in a raster fashion starting at the upper-left corner with index 0 and finishing at the lower-right corner with index 8 [see Fig. 6(b)]. For the four-tap OBMC filtering, the filter coefficient indices for a pixel in the second quadrant of the 16×16 block are shown in Fig. 4(a). The OBMC filter coefficient indexing for pixels in all

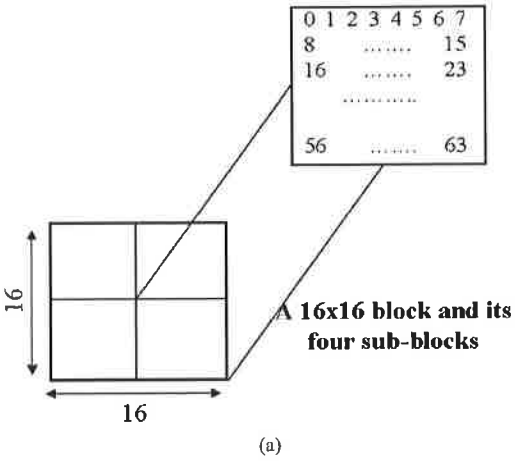


Fig. 6. LF indexing. Indexing of: (a) the 64 loop filters and (b) a 3×3 loop filter.

the four quadrants are given in Fig. 4(b), where a star indicates the current quadrant.

First, consider configuration 1. Let $\hat{I}_k$ and $\tilde{I}_k$ represent the predicted and reconstructed kth frame, respectively, and $e_k = I_k - \hat{I}_k$ and $\hat{e}_k$ represent the residual signal and its quantized counterpart, respectively. The objective function to minimize is given by

$$f(G,H) = E\left\{\left\{I_k(s) - \sum_{i=0}^{3} g(\check{s},i)\left[\hat{I}_{k-1}\left(s+v_i^s\right) + \sum_{j=0}^{8} h\left(\overline{s+v_i^s},j\right)\hat{e}_{k-1}\left(s+v_i^s+n_j\right)\right]\right\}^2\right\}$$

where v_i^s is the ith neighboring motion vector of pixel s, n_j belongs to the Kronecker product of $\{-1,0,1\}$ with itself, $\overline{s+v_i^s} = (s+v_i^s) \bmod 8$, and if $r^* = r \bmod 16$ (similarly for c), then

$$\check{r} = \begin{cases} r, & r^* < 8 \\ 15 - r, & \text{o.w.} \end{cases}$$

and $\check{s} = (\check{r}, \check{c})$. In the objective function $f(G,H), G = [g_0^T \; g_1^T \; \cdots \; g_{63}^T]^T$ and $H = [h_0^T \; h_1^T \; \cdots \; h_{63}^T]^T$.

We take an iterative approach to obtain a (local) optimal solution. In each iteration, we first fix $\{h_i\}$, and optimize OBMC filters $\{g_i\}$; then we fix $\{g_i\}$ and optimize loop filters $\{h_i\}$. Such

a procedure is iterated, until convergence is achieved. In our experiments, the convergence is typically obtained after three or four iterations.

When loop filters $\{h_i\}$ are fixed, we can define $\bar{e}_k$ as the result of applying LF on $\hat{e}_k$

$$\bar{e}_k(s) = \sum_{j=0}^{8} h(\bar{s}, j)\hat{e}_k(s + n_j) \tag{5}$$

and define $\bar{I}_k(s) = \hat{I}_k(s) + \bar{e}_k(s)$. Then the objective function becomes

$$\bar{f}(G) = E\left\{\left[I_k(s) - \sum_{i=0}^{3} g(\check{s}, i)\bar{I}_{k-1}\left(s + v_i^s\right)\right]^2\right\} \tag{6}$$

whose solution is given by

$$g_i = r_i^{-1}\left[\psi_i - u_4\left(\frac{u_4^T r_i^{-1}\psi_i - 1}{u_4^T r_i^{-1} u_4}\right)\right] \tag{7}$$

where r_i and ψ_i are the ith auto-correlation and cross-correlation matrices, respectively. With fixed OBMC filters $\{g_i\}$, we optimize $\{h_i\}$ by minimizing the quantization error $E[e_k(s) - \hat{e}_k(s)]^2$, the solution of which is easy to derive.

As to configuration 2, the objective function to minimize is given by

$$f(G,H) = E\left\{\left\{I_k(s) - \sum_{i=0}^{3} g(\check{s}, i)\sum_{j=0}^{8} h\left(\overline{s + v_i^s}, j\right) \times \left[\hat{I}_{k-1}\left(s + v_i^s + n_j\right) + \hat{e}_{k-1}\left(s + v_i^s + n_j\right)\right]\right\}^2\right\}.$$

The iterative method is also applied to obtain an optimal solution. When the loop filters are fixed, the same approach as used above can be employed to obtain optimal OBMC filters. With fixed OBMC filters, optimal $\{h_i\}$ are found by minimizing $E[(I_k(s) - \hat{I}_k(s) - \hat{e}_k(s))^2]$.

For configuration 3, we design the linear filters by minimizing

$$f(G,H) = E\left\{\left[I_k(s) - \sum_{i=0}^{3} g(\check{s}, i)\sum_{j=0}^{8} h\left(\overline{s + v_i^s}, j\right) \times \check{I}_{k-1}\left(s + v_i^s + n_j\right)\right]^2\right\}.$$

Again, we use the iterative method to look for a local optimal solution. With fixed loop filters $\{h_i\}$, OBMC filters are computed in the same way as for configurations 1 and 2. However, when OBMC filters $\{g_i\}$ are fixed, there is no simplification for the objective function $f(G,H)$ above. By using the Lagrangian multiplier method, we have

$$\nabla_H\left(f(G,H) + \sum_{i=0}^{63}\lambda_i h_i^T u_9\right) = RH - \Psi + \begin{bmatrix}\lambda_0 u_9 \\ \vdots \\ \lambda_{63} u_9\end{bmatrix} = 0 \tag{8}$$

where λ_i is the ith Lagrange multiplier, and

$$H = R^{-1}[\Psi - U^T(UR^{-1}U^T)^{-1}(UR^{-1}\Psi) - u_{64}]$$

where

$$U(i,j) = \begin{cases} 1, & \text{if } j = \lfloor i/9 \rfloor \\ 0, & \text{o.w.} \end{cases}$$

$\lfloor\cdot\rfloor$ is the floor integer operator, R and Ψ are the auto- and cross-correlation matrices, respectively, and $H^T = [h_0^T\ h_1^T\ \ldots\ h_{63}^T]$.

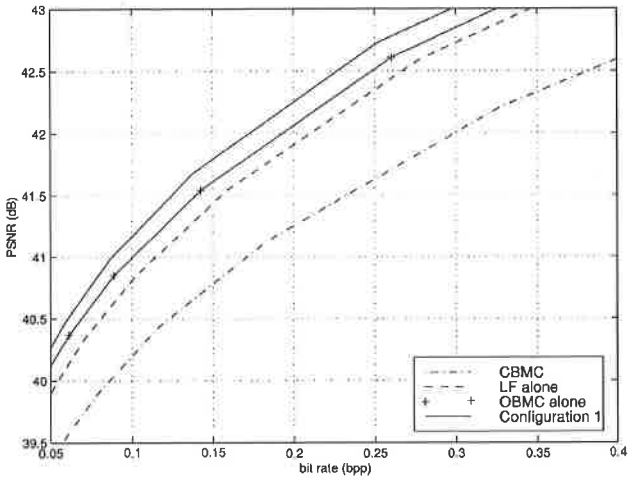


Fig. 7. Rate-distortion performance comparison using Miss America.

V. Experimental Results

We use 60-frame CIF sequences Miss America and Football for experiments. The filters are trained using a group of other CIF sequences including Salesman, Tennis, and Claire. Each training sequence is ten-frame long. The rate-distortion performance of all the techniques mentioned above is compared in the following, where rate is given as the number of bits used to encode each pixel averaged over the whole sequence and distortion is given as the average PSNR. Full-search translational block motion estimation with Euclidean distance are used in the experiments except where it is noted.

Shown in Fig. 7 are the rate-distortion curves for the video coder using conventional block motion compensation (CBMC), OBMC alone, LF alone, and system configuration 1. The sequence is Miss America. The same block motion estimator is used in all cases. When used alone, OBMC outperforms LF by 0.2 dB, while when they are jointly applied in the form of configuration 1, another 0.2-dB improvement is achieved. In Figs. 8 and 9, all three proposed configurations are compared, together with the case of using OBMC only. It is seen that configurations

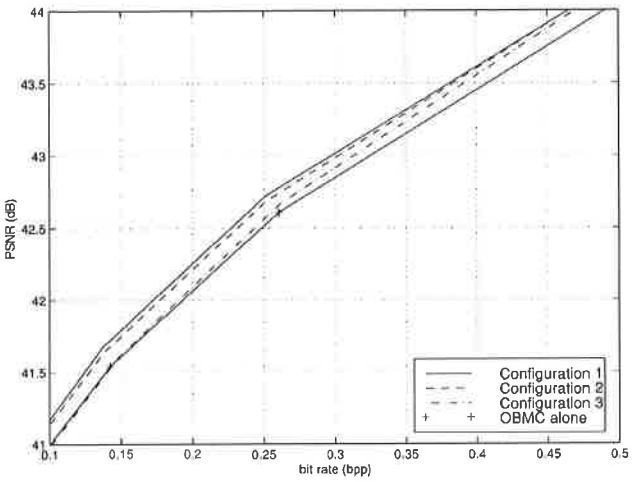


Fig. 8. Comparison of different configurations at relatively high bit rates using Miss America.

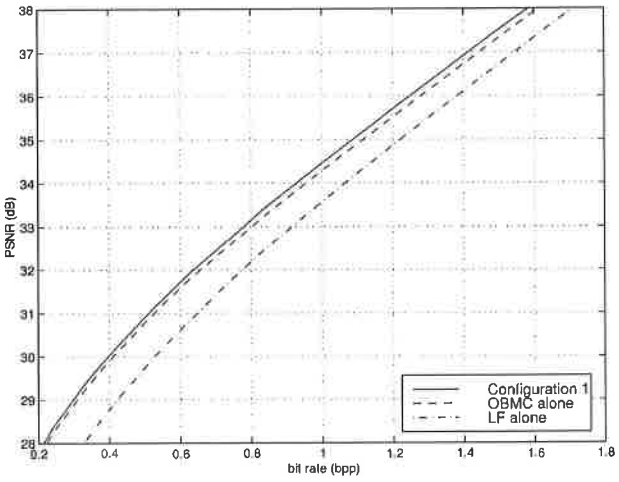


Fig. 10. Rate-distortion results using Football sequence.

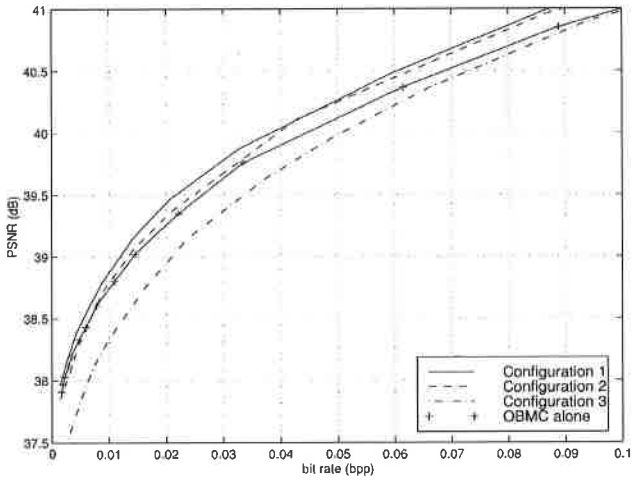


Fig. 9. Comparison of different configurations at relatively low bit rates using Miss America.

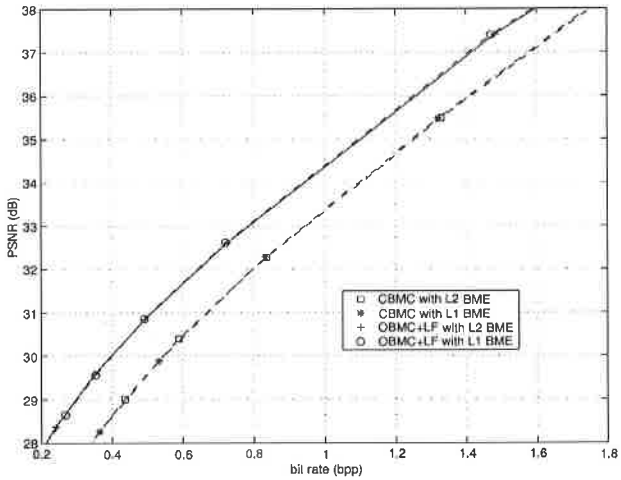


Fig. 11. Rate-distortion results using both l_1 and l_2 block motion estimation for Football sequence.

1 and 2 consistently beat the single use of OBMC, with configuration 1 slightly edging out configuration 2, while configuration 3 only helps when the bit rate is relatively high.

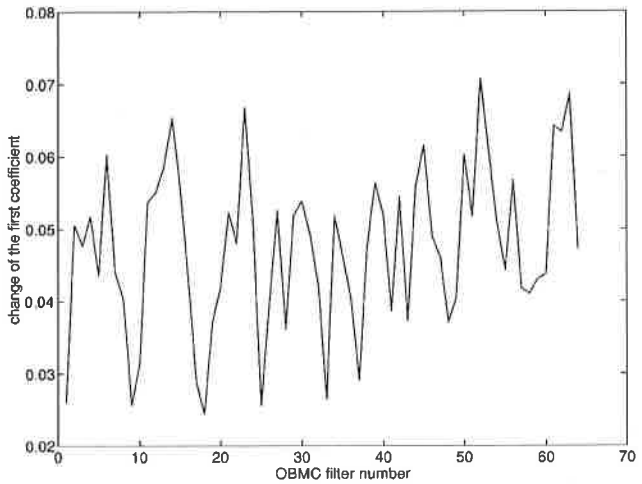
Shown in Fig. 10 are the rate-distortion curves obtained using 60-frame Football sequence, for OBMC alone, LF alone, and system configuration 1. It is worth noting that, although this sequence is considerably different from the Miss America sequence, the gain by joint application of OBMC and LF is still 0.2 dB at virtually all bit rates.

The experiments confirm that system configuration 1 is the best among the three configurations considered. This is not unexpected, since the quantization noise is generated when the residual signal is quantized, while as shown in Fig. 5, the LF component of configuration 1 filters the quantized residual signal directly.

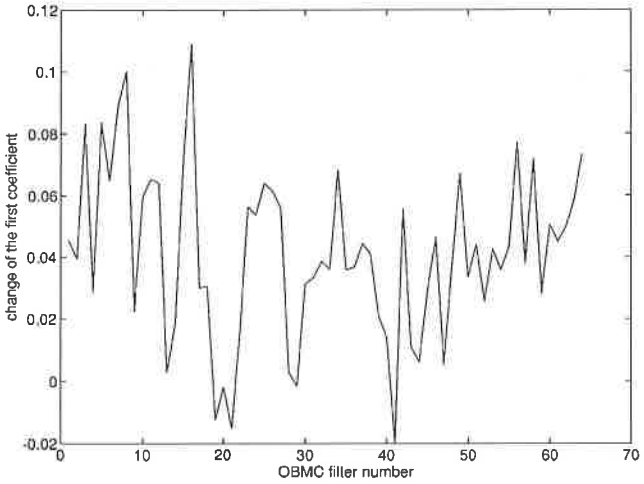
We have used Euclidean-distance motion estimation in the analysis in Section II, as well as in the experiments above. As mentioned in Section II, we believe that the results are quite general for other translational block motion estimation schemes. This is verified by results given in Fig. 11, where the rate-distortion performance using both l_1 and l_2 block motion estimation and both OBMC + LF and CBMC are given. There is very little difference between the two motion estimation procedures.

Next, we inspect how loop filters and OBMC filters change, going from stand-alone use to joint application with each other. Through such comparisons, one can get more knowledge about how the (optimal) OBMC filters and loop filters relate to motion uncertainty and quantization noise. The optimal filters in configuration 1 as obtained in the previous section will be used for comparison. Recall that both OBMC filters and loop filters are constrained to have unit gain.

Let us look at the OBMC filters, especially the first OBMC coefficient that corresponds to the motion estimate of the current block (i.e., the first element of each 4×1 vector g_i). It has the largest weight among all four coefficients. We first consider the situation without LF. The optimal OBMC filters are computed with two different noise statistics. In one case, the reconstructed previous signal is corrupted by quantization noise,

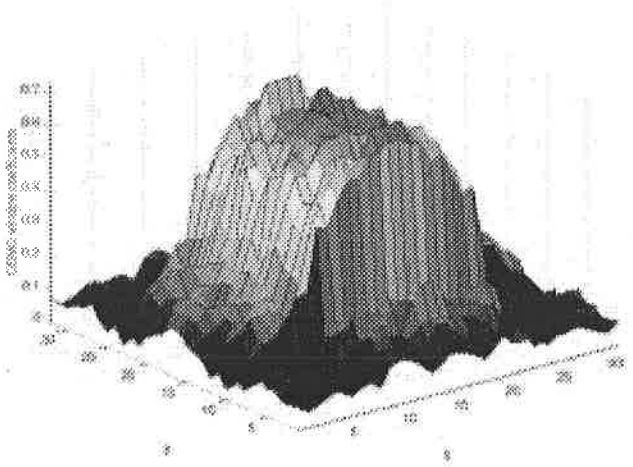


(a)

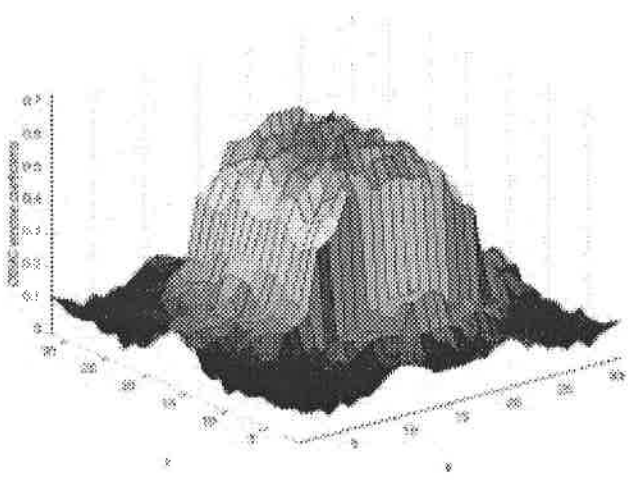


(b)

Fig. 12. Change of the first OBMC filter coefficient from: (a) predicting with quantization noise-corrupted previous frame to predicting with noise-free previous frame and (b) stand-alone use to joint application with LF.



(a)



(b)

Fig. 13. Optimal OBMC windows when applied: (a) without LF and (b) with LF. The window in (a) is flatter because of the need to suppress both motion uncertainty and quantization nosie.

while in the other case, it is noise free (no quantization takes place). Shown in Fig. 12(a) is the change of the first OBMC coefficient. It is seen that when the previous frame is noisy, the OBMC filters are more spread out in order to deal with both motion uncertainty and quantization noise, while when the noise is absent, the OBMC filters concentrate more weight on the first coefficient, which reflects greater confidence in the motion estimate for the current block. Shown in Fig. 12(b) is the change of the first OBMC filter coefficient, going from stand-alone use to joint use with loop filters. This can be understood the same as the previous comparison. Effective LF reduces quantization noise, and the OBMC estimator can place more confidence on the motion estimate of the current block. The corresponding OBMC windows are shown in Fig. 13. It is seen that the window applied alone in Fig. 13(a) is flatter than the window applied together with LF.

Next, we look at the loop filters. Shown in Fig. 14(a) are the coefficient values for loop filters used alone and when jointly applied with OBMC. The change of the middle coefficient is given in Fig. 14(b). It is seen that when loop filters are jointly applied with overlapped windows, the filters tend to place more weight on the central coefficient. The intuitive explanation is that the existence of motion uncertainty requires the use of several motion estimates to obtain good prediction. Therefore, both quantization noise suppression and motion uncertainty reduction require to spread the weight among all coefficients. Without OBMC, loop filters have to adjust to both requirements, and the weights are spread out. After OBMC is turned on, the loop filters are "freed" from taking care of multiple motion possibilities, and thus the weights are more concentrated. Through the changes occurred to OBMC filters and loop filters, it is confirmed that our objective of suppressing both noise sources has been achieved.

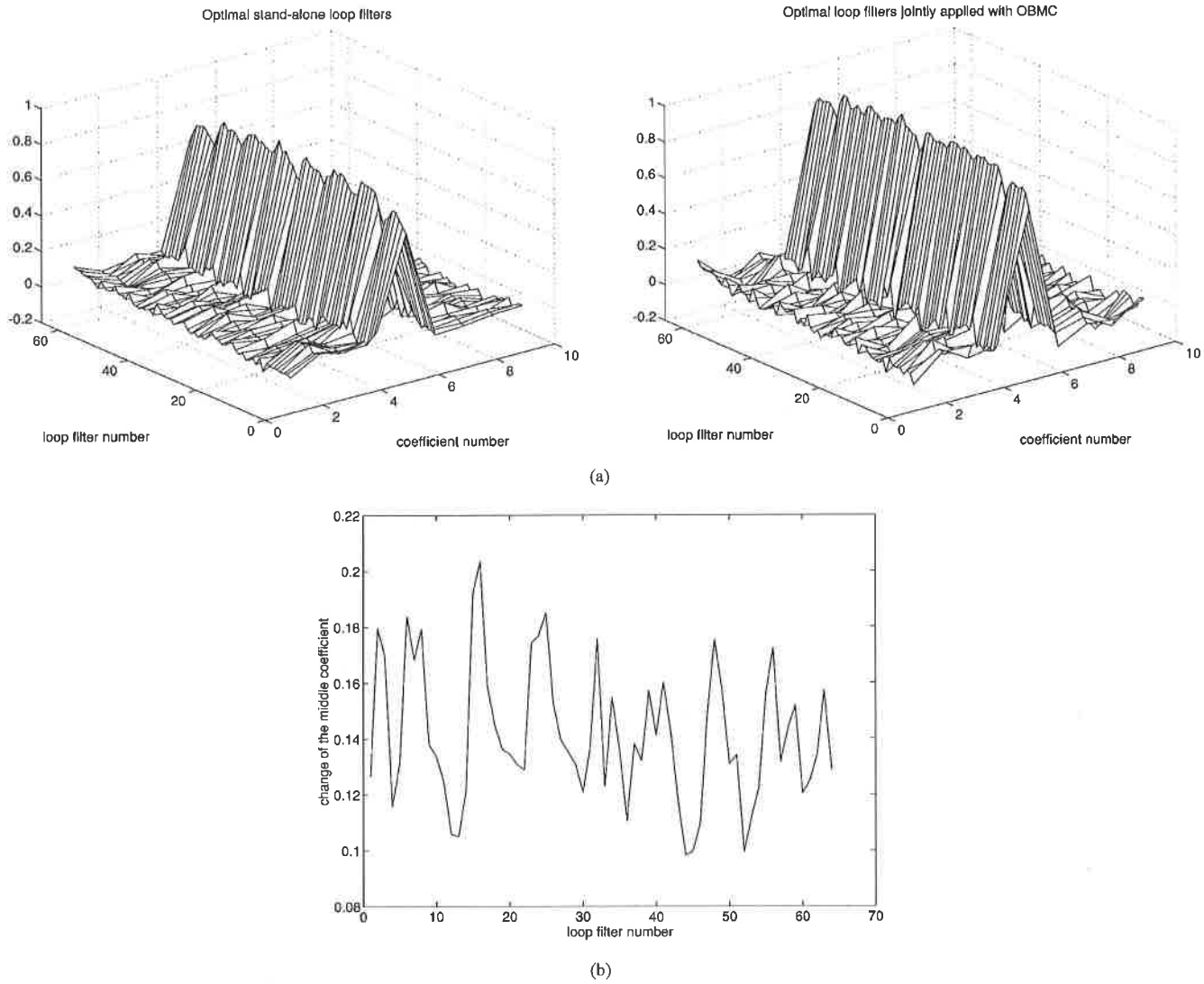


Fig. 14. (a) Comparison of loop filters applied stand-alone and loop filters jointly applied with OBMC. (b) Change of the middle loop filter coefficient.

As the final examination, we give reconstructed pictures in Fig. 15. Both pictures are the same frame (frame 10) in Football sequence and are coded using similar number of bits. It is seen that compared with the picture generated using OBMC only, the picture produced by using OBMC + LF is smoother. For instance, it has less ringing noises because of the use of block-adaptive LF.

VI. Conclusion

In this work, we have modeled motion uncertainty and quantization noise in a hybrid video coder as two distinctive noise sources corrupting the motion compensation performance, and analyzed their block-structured nonstationarity. A good motion compensation algorithm has to take into consideration both factors to maximize prediction gain. We have proposed to jointly apply overlapped block motion compensation and LF at the same time, taking advantage of both the ability of OBMC to reduce motion uncertainty and the ability of LF to mitigate quantization noise. Algorithms to design optimal space-varying OBMC filters and loop filters are presented. Better rate-distortion performance is achieved at virtually all bit rates.

As to future works, besides pursuing the rate-distortion performance, it is also worth investigating what filters are best suited for improving the subjective quality. This is especially the case with the LF component in the system. Furthermore, nonlinear filtering framework such as a neural network-based predictor, which accounts for both motion uncertainty and quantization noise, may be an interesting alternate to the linear filtering approach. It is also of great interests to find out how our approach compares to the deblocking filter introduced to ITU-T H.263+ [9], which came to our awareness after we had started this work.

The tests in this work have used integer motion estimation exclusively. On the other hand, it is well known that sub-pixel motion estimation have somewhat similar effects as LF [10]. In fact, for a given motion compensation scheme with OBMC

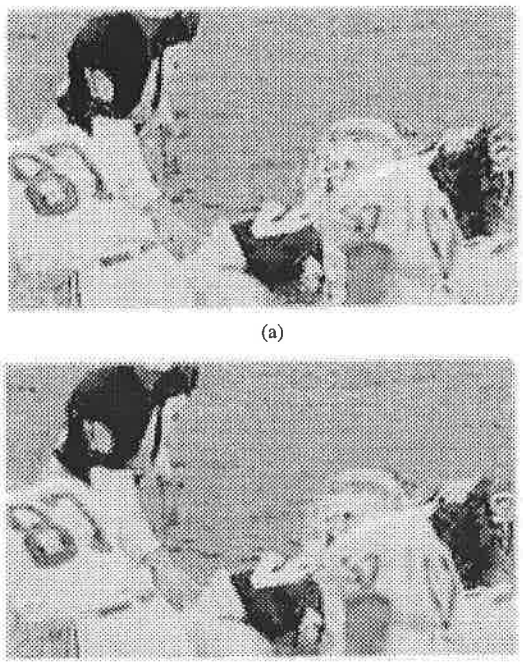


Fig. 15. Coding results: (a) OBMC only and (b) OBMC + LF.

and loop filter, (sub)optimal motion estimation can be computed [6], [11]. It is interesting to know how much gain can be achieved with sub-pixel search, and perhaps even more interesting to jointly optimize motion estimation and the proposed OBMC + LF technique.

ACKNOWLEDGMENT

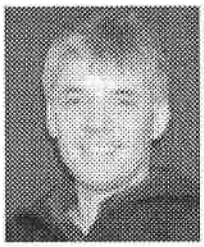
The authors thank anonymous reviewers for their helpful comments that made the paper better.

REFERENCES

[1] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE J. Select. Areas Commun.*, vol. 5, pp. 1140–1154, 1987.
[2] K. Pang and T. Tan, "Optimum loop filter in hybrid coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 158–167, Feb. 1994.
[3] H. Watanabe and S. Singhal, "Windowed motion compensation," in *Proc. Visual Communication and Image Processing—SPIE*, vol. 1605, 1991.
[4] S. Nogaki and M. Ohta, "An overlapped block motion compensation for high quality motion picture coding," in *Proc. Int. Symp. Circuits and Systems*, 1992.
[5] *Recommendation H.263, Video Coding for Low Bitrate Communication*, International Telecommunication Union, 1995.
[6] M. Orchard and G. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *IEEE Trans. Image Processing*, vol. 3, pp. 693–699, 1994.
[7] B. Tao, M. Orchard, and B. Dickinson, "Joint application of overlapped block motion compensation and loop filtering for low bit-rate video coding," in *Proc. Int. Conf. Image Processing*, vol. III, 1997, pp. 626–629.
[8] B. Tao, "Optimization of Hybrid Video Coders," Ph.D. dissertation, Princeton Univ., Princeton, NJ, 1998.
[9] Recommendation H.263+ Draft, Video Coding for Low Bitrate Communication, 1998.
[10] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, pp. 604–611, Apr. 1993.
[11] B. Tao and M. Orchard, "Non-iterative motion estimation for overlapped block motion compensation," in *Proc. Visual Communications and Image Processing*, 1998, pp. 1032–1040.

Bo Tao (S'94–M'99) received the B.S. degree from Tsinghua University, China, in 1992, and the Ph.D. degree from Princeton University, Princeton, NJ, in 1998, both in electrical engineering.

From 1998 to 2000, he was with Sony U.S. Research Labs, San Jose, CA, as a Member of Technical Staff. He is now with Streaming21 Inc., Los Gatos, CA, as a Senior Research Staff member. His research interests include image and video processing and transmission. He has published more than 30 technical papers and has several patents pending on digital camera technologies.



Michael T. Orchard (F'00) was born in Shanghai, China, and grew up in New York. He received the B.S. and M.S. degrees in electrical engineering from San Diego State University, San Diego, CA, in 1980 and 1986, respectively, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1988 and 1990.

He was with the Government Products Division of Scientific Atlanta from 1982 to 1986, developing passive sonar DSP applications, and has consulted with the Visual Communications Department of AT&T Bell Laboratories since 1988. From 1990 to 1995, he was an Assistant Professor with the Department of Electrical and Computer Engineering, the University of Illinois at Urbana-Champaign, where he served as Associate Director of the Image Laboratory of the Beckman Institute. Since 1995, he has been an Associate Professor with the Department of Electrical Engineering, Princeton University. During the spring of 2000, he served as Texas Instruments Visiting Professor at Rice University.

Dr. Orchard received the National Science Foundation Young Investigator Award in 1993, the Army Research Office Young Investigator Award in 1996, and was elected IEEE Fellow in 2000 for "contributions to the theory and development of image and video compression algorithms."